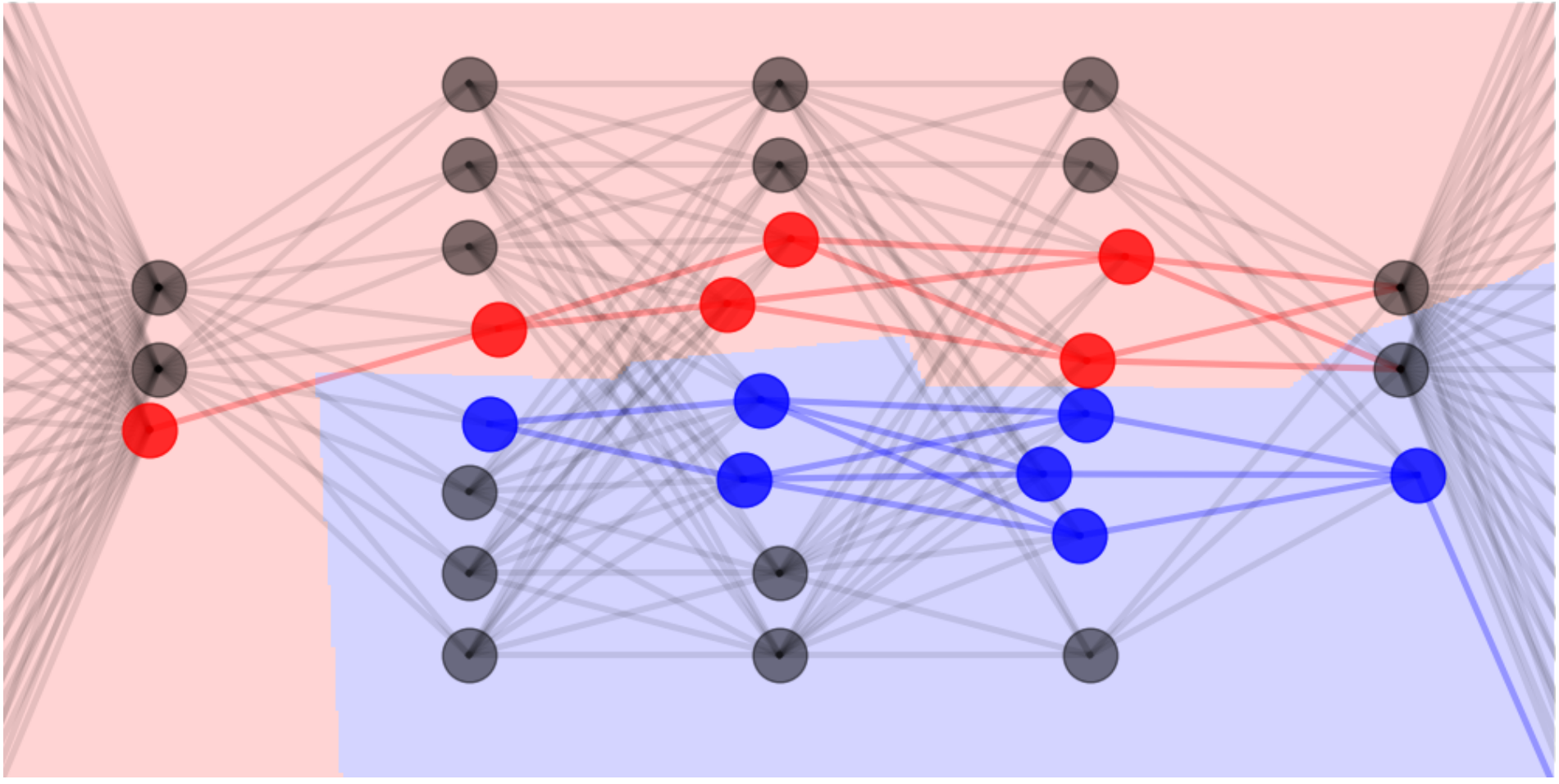
TU-Delft Machine and Deep Learning



Feed-forward networks

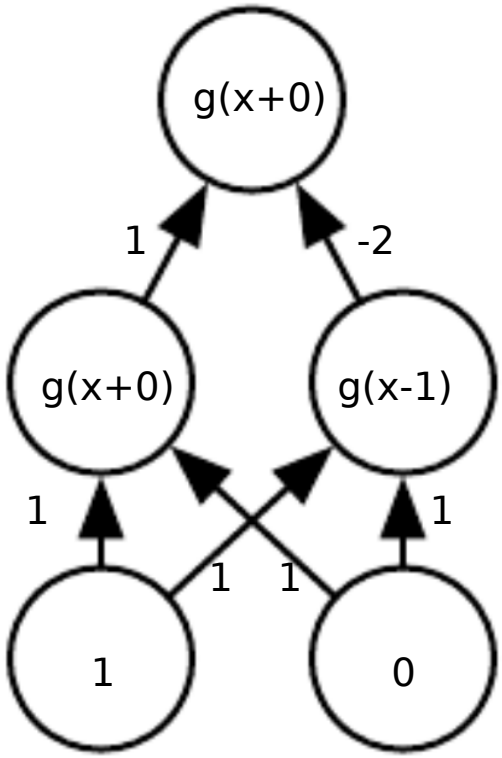
Lecture 1

DL-book: chapters 6, 6.1, 4.3, 5.9; UDL-book: chapters 3, 4.

Lecturer: Jan van Gemert

What is the outcome of this Feed-forward net?

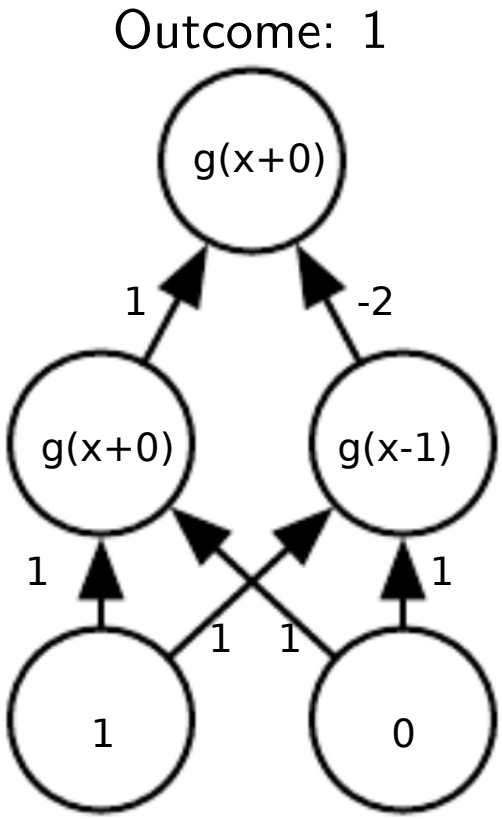
DL-book: 6, 6.1



$$g(z) = \max\{0, z\}$$

What is the outcome of this Feed-forward net?

DL-book: 6, 6.1

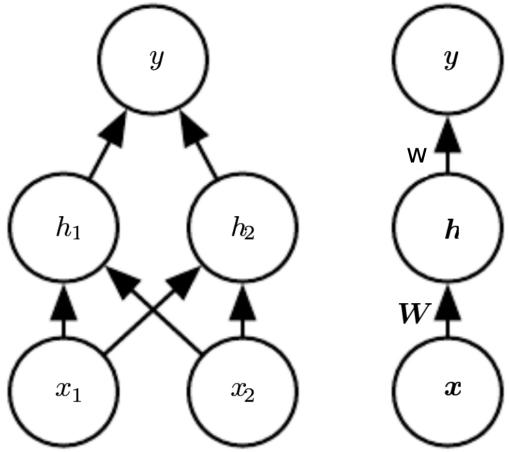


$$g(z) = \max\{0, z\}$$

Deep feed-forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1, 4.3, 5.9

- Approximate some function f^*
- Eg: classifier $y = f^*(\mathbf{x})$
- Learn parameters θ for mapping $y = f(\mathbf{x}; \theta)$

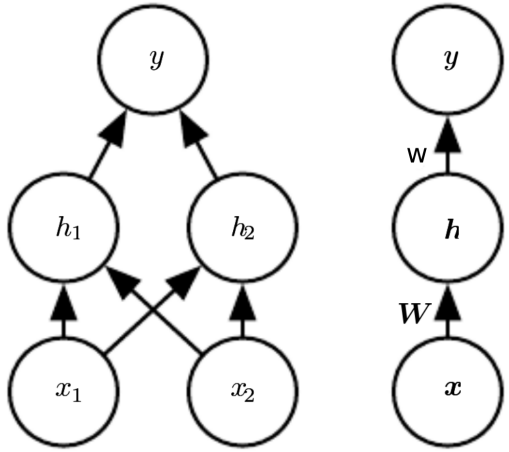


Deep feed-forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1, 4.3, 5.9

- Approximate some function f^*
- Eg: classifier $y = f^*(\mathbf{x})$
- Learn parameters θ for mapping $y = f(\mathbf{x}; \theta)$

Q: How many parameters (length of θ) ?

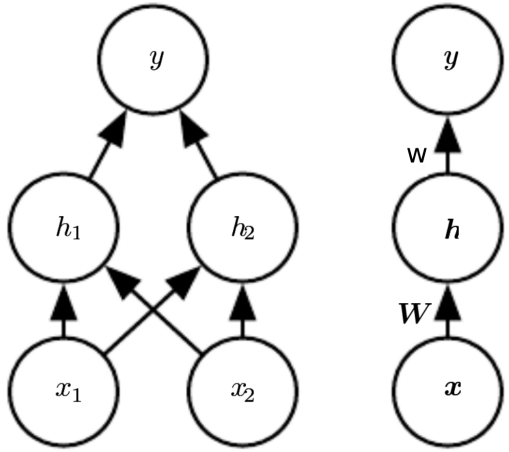


Deep feed-forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1, 4.3, 5.9

- Approximate some function f^*
- Eg: classifier $y = f^*(\mathbf{x})$
- Learn parameters θ for mapping $y = f(\mathbf{x}; \theta)$

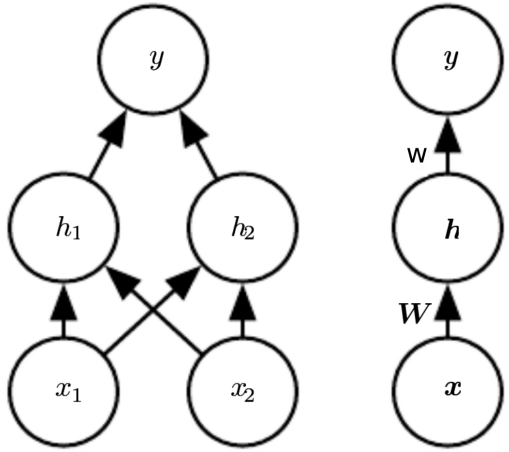
Q: How many parameters (length of θ) ? A: 9



Deep feed-forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1, 4.3, 5.9

- Approximate some function f^*
- Eg: classifier $y = f^*(\mathbf{x})$
- Learn parameters θ for mapping $y = f(\mathbf{x}; \theta)$

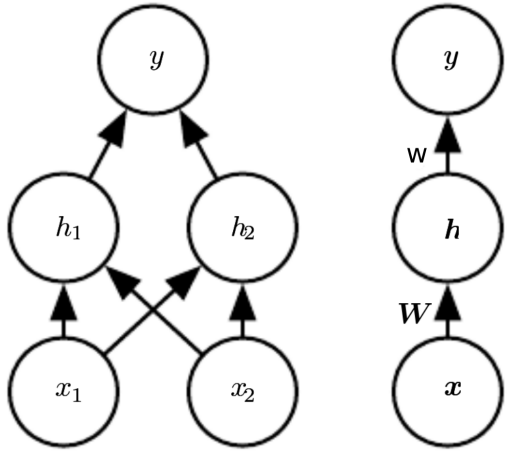


Q: How many parameters (length of θ) ? A: 9
Q: Are the two graphs the same?

Deep feed-forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1, 4.3, 5.9

- Approximate some function f^*
- Eg: classifier $y = f^*(\mathbf{x})$
- Learn parameters θ for mapping $y = f(\mathbf{x}; \theta)$



Q: How many parameters (length of θ) ? A: 9

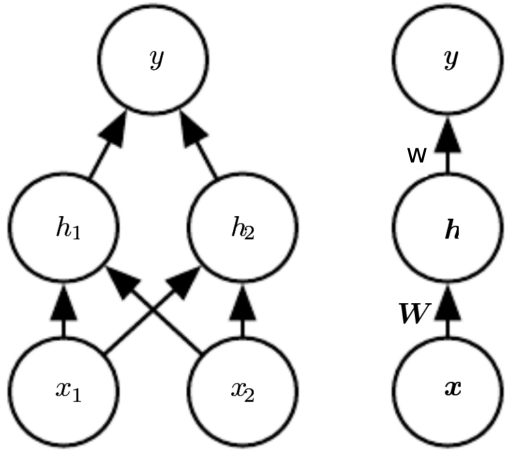
Q: Are the two graphs the same?

A: $g(x_1 w_1 + x_2 w_2 + c_1) w_5 + g(x_1 w_3 + x_2 w_4 + c_2) w_6 + b = g(\mathbf{x}^\top \mathbf{W} + \mathbf{c}^\top)\mathbf{w} + b$

Deep feed-forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1, 4.3, 5.9

- Approximate some function f^*
- Eg: classifier $y = f^*(\mathbf{x})$
- Learn parameters θ for mapping $y = f(\mathbf{x}; \theta)$



Q: How many parameters (length of θ) ? A: 9

Q: Are the two graphs the same?

A: $g(x_1 w_1 + x_2 w_2 + c_1) w_5 + g(x_1 w_3 + x_2 w_4 + c_2) w_6 + b = g(\mathbf{x}^\top \mathbf{W} + \mathbf{c}^\top)\mathbf{w} + b$

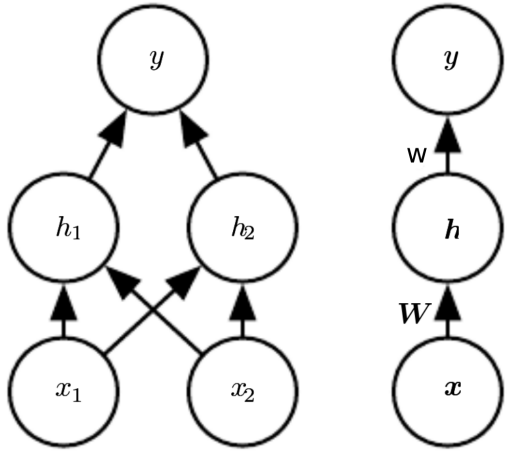
Networked in a function chain $f(\mathbf{x}) = f^{(1)}(f^{(2)}(f^{(3)}(\mathbf{x})))$

Q: For f, what is: first layer? Second? Output? Hidden? Depth?

Deep feed-forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1, 4.3, 5.9

- Approximate some function f^*
- Eg: classifier $y = f^*(\mathbf{x})$
- Learn parameters θ for mapping $y = f(\mathbf{x}; \theta)$



Q: How many parameters (length of θ) ? A: 9

Q: Are the two graphs the same?

A: $g(x_1 w_1 + x_2 w_2 + c_1) w_5 + g(x_1 w_3 + x_2 w_4 + c_2) w_6 + b = g(\mathbf{x}^\top \mathbf{W} + \mathbf{c}^\top)\mathbf{w} + b$

Networked in a function chain $f(\mathbf{x}) = f^{(1)}(f^{(2)}(f^{(3)}(\mathbf{x})))$

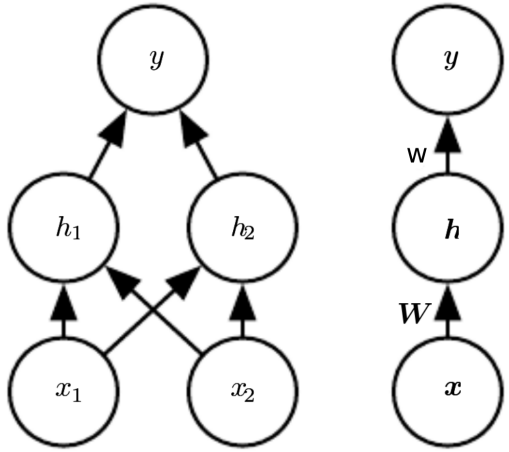
Q: For f, what is: first layer? Second? Output? Hidden? Depth?

A: first layer: $f^{(3)}$, Second: $f^{(2)}$, Output: $f^{(1)}$, Hidden: $f^{(2)}$, Depth: 3

Deep feed-forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1, 4.3, 5.9

- Approximate some function f^*
- Eg: classifier $y = f^*(\mathbf{x})$
- Learn parameters θ for mapping $y = f(\mathbf{x}; \theta)$



Q: How many parameters (length of θ) ? A: 9
Q: Are the two graphs the same?
A: $g(x_1 w_1 + x_2 w_2 + c_1) w_5 + g(x_1 w_3 + x_2 w_4 + c_2) w_6 + b = g(\mathbf{x}^\top \mathbf{W} + \mathbf{c}^\top)\mathbf{w} + b$

Networked in a function chain $f(\mathbf{x}) = f^{(1)}(f^{(2)}(f^{(3)}(\mathbf{x})))$

Q: For f, what is: first layer? Second? Output? Hidden? Depth?
A: first layer: $f^{(3)}$, Second: $f^{(2)}$, Output: $f^{(1)}$, Hidden: $f^{(2)}$, Depth: 3

Goal: set θ so that $f(\mathbf{x}; \theta)$ matches $f^*(\mathbf{x})$

Training a network

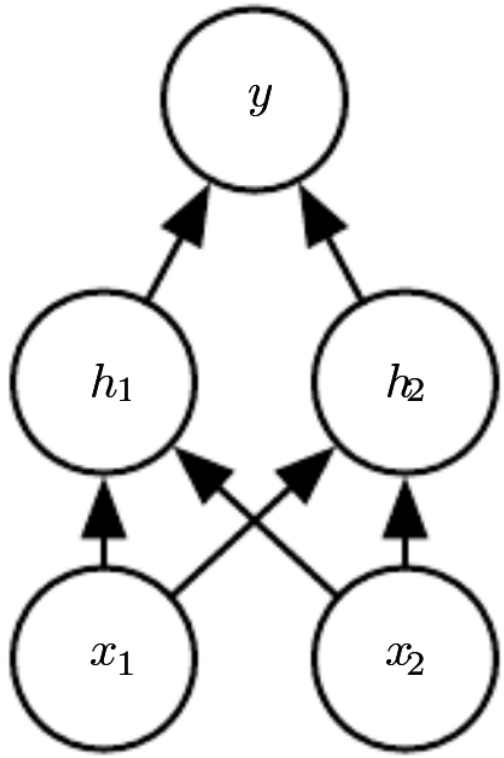
DL-book: 4.3

Training set:

Data		Label
0	1	1
1	1	0
0	0	0
1	0	1

While not converged:

1. Present a training sample
2. Compare the results
3. Update the parameters



Training a network

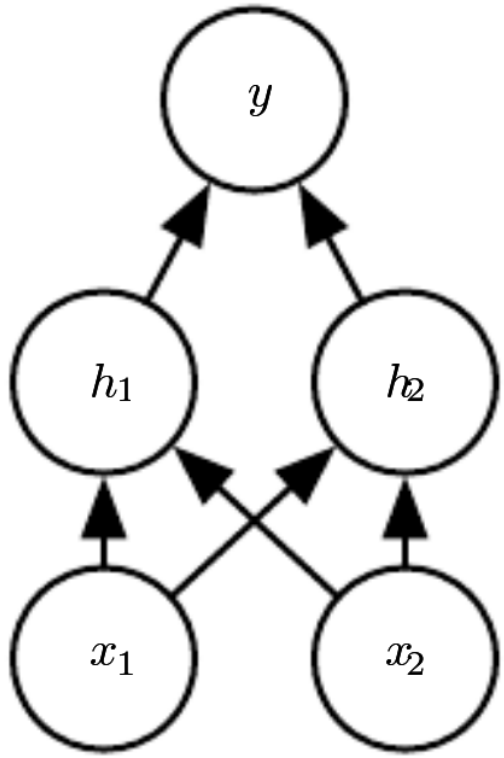
DL-book: 4.3

Training set:

Data		Label
0	1	1
1	1	0
0	0	0
1	0	1

While not converged:

1. Present a training sample
2. Compare the results
3. Update the parameters



Lets do a few steps of this algorithm... Q: What is the first step?

Training a network

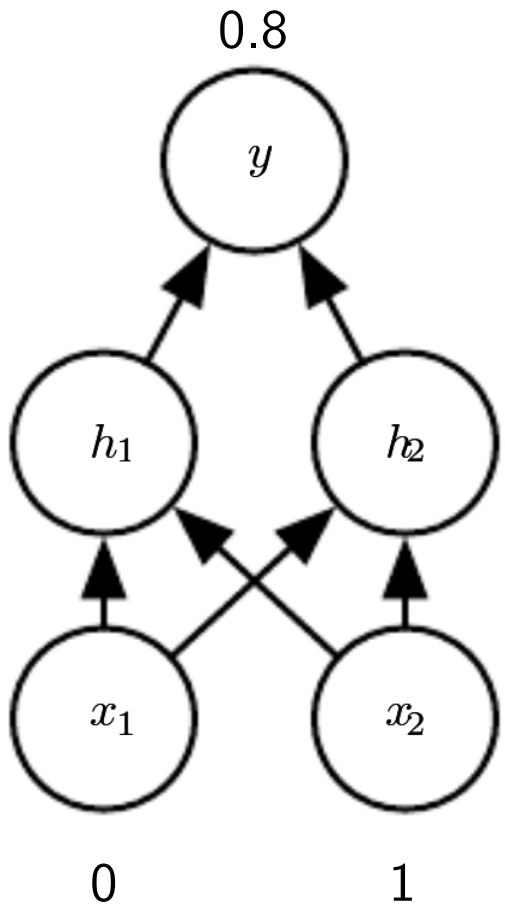
DL-book: 4.3

Training set:

Data		Label
0	1	1
1	1	0
0	0	0
1	0	1

While not converged:

1. Present a training sample
2. Compare the results
3. Update the parameters



Training a network

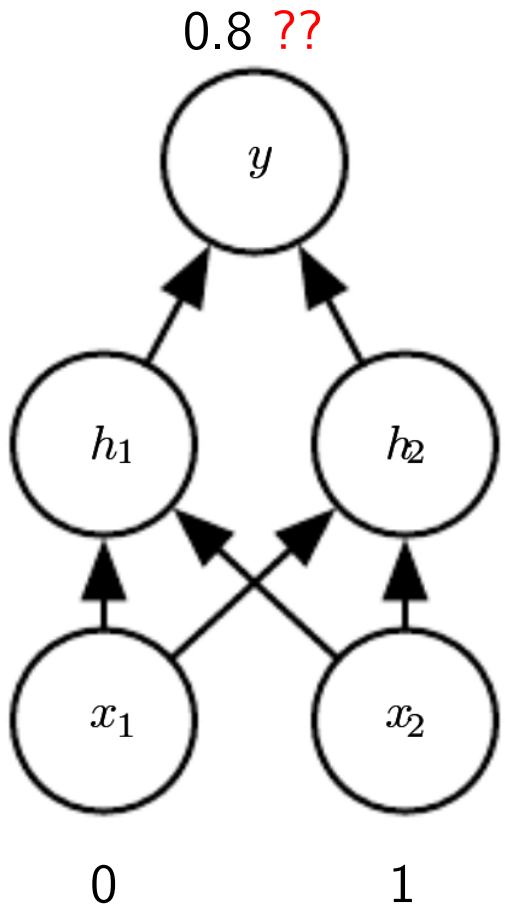
DL-book: 4.3

Training set:

Data		Label
0	1	1
1	1	0
0	0	0
1	0	1

While not converged:

1. Present a training sample
2. Compare the results
3. Update the parameters



Training a network

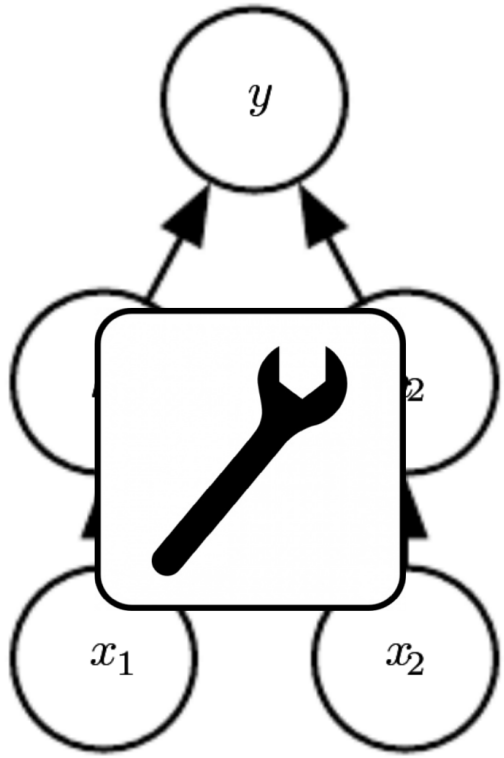
DL-book: 4.3

Training set:

Data		Label
0	1	1
1	1	0
0	0	0
1	0	1

While not converged:

1. Present a training sample
2. Compare the results
3. Update the parameters



Training a network

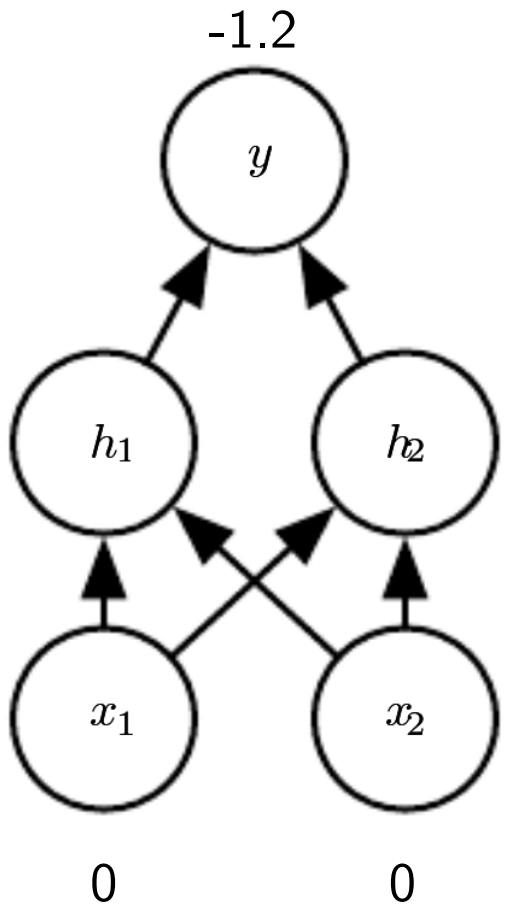
DL-book: 4.3

Training set:

Data		Label
0	1	1
1	1	0
0	0	0
1	0	1

While not converged:

1. Present a training sample
2. Compare the results
3. Update the parameters



Training a network

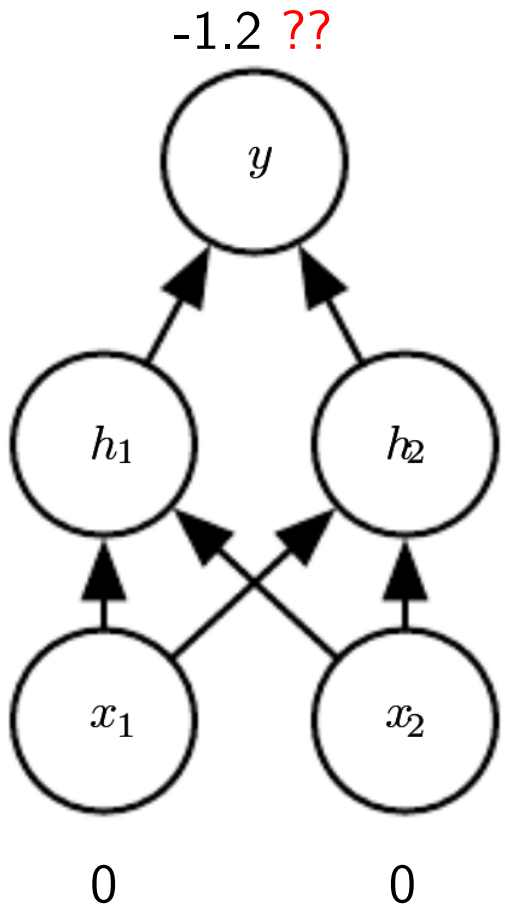
DL-book: 4.3

Training set:

Data		Label
0	1	1
1	1	0
0	0	0
1	0	1

While not converged:

1. Present a training sample
2. Compare the results
3. Update the parameters



Training a network

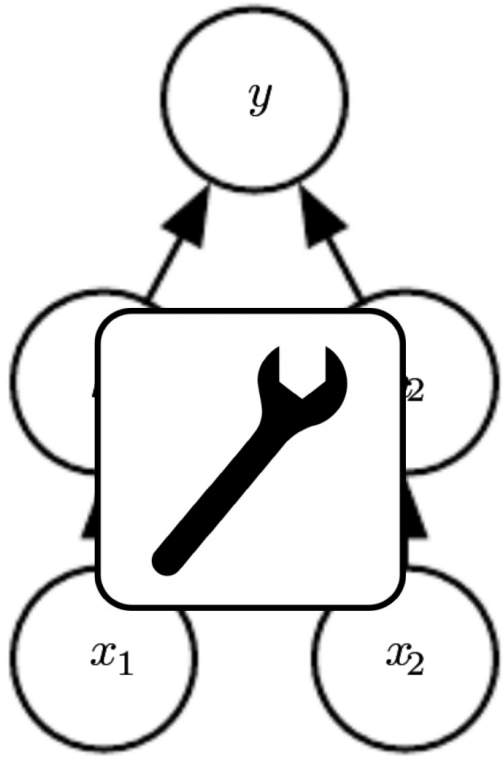
DL-book: 4.3

Training set:

Data		Label
0	1	1
1	1	0
0	0	0
1	0	1

While not converged:

1. Present a training sample
2. Compare the results
3. Update the parameters



Questions?

Update parameters

DL-book: 4.3

- Minimize some criterion, an objective/cost/loss/error-function
  For example, the squared difference between the outcome and the label.

Update parameters

DL-book: 4.3

- Minimize some criterion, an objective/cost/loss/error-function
  For example, the squared difference between the outcome and the label.
- Q: For $f(w)$ in 1d, If I make a small change to w, how does it change $f(w)$?

Update parameters

DL-book: 4.3

- Minimize some criterion, an objective/cost/loss/error-function
  For example, the squared difference between the outcome and the label.
- Q: For $f(w)$ in 1d, If I make a small change to w, how does it change $f(w)$?
  A: Derivative f' or $\frac{\partial f}{\partial w}$, tells you $f(w+\epsilon) \approx f(w) + \epsilon f'(w)$

Update parameters

DL-book: 4.3

- Minimize some criterion, an objective/cost/loss/error-function
  For example, the squared difference between the outcome and the label.
- Q: For $f(w)$ in 1d, If I make a small change to w, how does it change $f(w)$?
  A: Derivative f' or $\frac{\partial f}{\partial w}$, tells you $f(w+\epsilon) \approx f(w) + \epsilon f'(w)$
- Q: How to reduce loss?

Update parameters

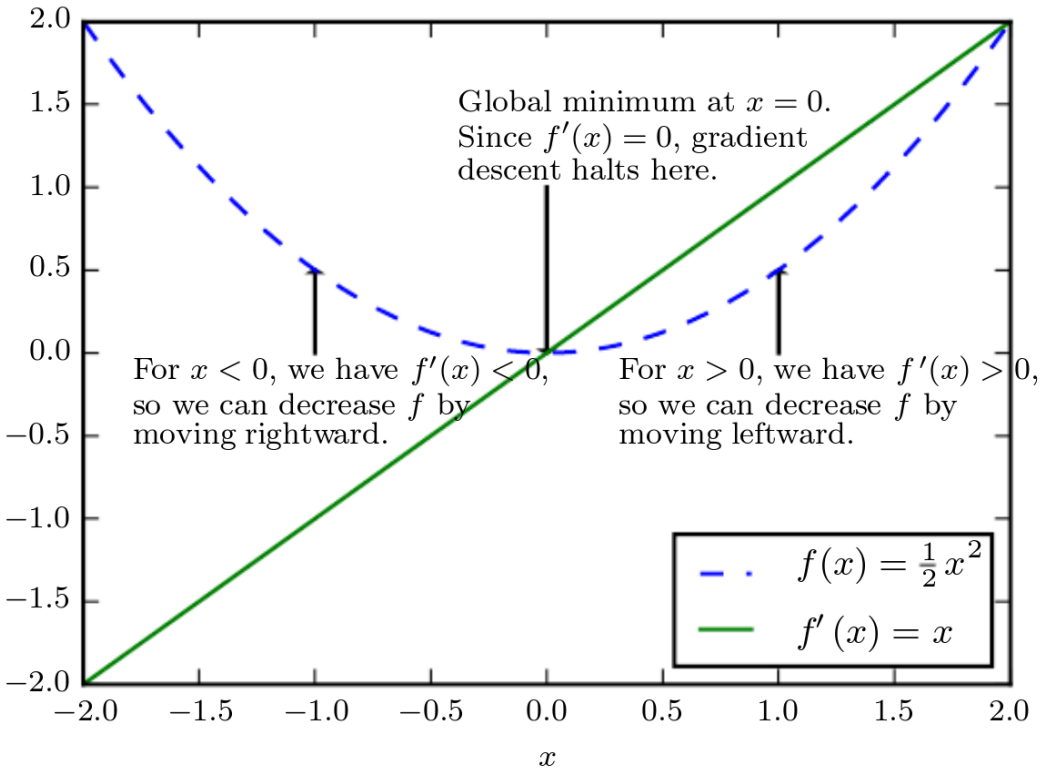
DL-book: 4.3

- Minimize some criterion, an objective/cost/loss/error-function
  For example, the squared difference between the outcome and the label.
- Q: For $f(w)$ in 1d, If I make a small change to w, how does it change $f(w)$?
  A: Derivative f' or $\frac{\partial f}{\partial w}$, tells you $f(w + \epsilon) \approx f(w) + \epsilon f'(w)$
- Q: How to reduce loss? A: By moving in the opposite sign of it's derivative.

Update parameters

DL-book: 4.3

- Minimize some criterion, an objective/cost/loss/error-function
  For example, the squared difference between the outcome and the label.
- Q: For $f(w)$ in 1d, If I make a small change to w, how does it change $f(w)$?
  A: Derivative f' or $\frac{\partial f}{\partial w}$, tells you $f(w+\epsilon) \approx f(w) + \epsilon f'(w)$
- Q: How to reduce loss? A: By moving in the opposite sign of it's derivative.



Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- Let the loss/error/cost function L be given by $L(\mathbf{X}; \mathbf{y}; \theta)$

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- Let the loss/error/cost function L be given by $L(\mathbf{X}; \mathbf{y}; \theta)$
- Q: What do the symbols mean?

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- Let the loss/error/cost function L be given by $L(\mathbf{X}; \mathbf{y}; \theta)$
- Q: What do the symbols mean?
  A: $\mathbf{X}$: data matrix; θ: learnable parameters; $\mathbf{y}$: ground truth vector;

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- Let the loss/error/cost function L be given by $L(\mathbf{X}; \mathbf{y}; \theta)$
- Q: What do the symbols mean?
  A: $\mathbf{X}$: data matrix; θ: learnable parameters; $\mathbf{y}$: ground truth vector;
- An example of loss/error/cost function: mean square error
  $L(\mathbf{X}, \mathbf{y}; \theta) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}^{(i)}; \theta) - y^{(i)})^2$

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- Let the loss/error/cost function L be given by $L(\mathbf{X}; \mathbf{y}; \theta)$
- Q: What do the symbols mean?
  A: $\mathbf{X}$: data matrix; θ: learnable parameters; $\mathbf{y}$: ground truth vector;
- An example of loss/error/cost function: mean square error
  $L(\mathbf{X}, \mathbf{y}; \theta) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}^{(i)}; \theta) - y^{(i)})^2$
- Q: What does a partial derivative $\frac{\partial}{\partial \theta_k} L(\mathbf{X}; \mathbf{y}; \theta)$ represent?

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- Let the loss/error/cost function L be given by $L(\mathbf{X}; \mathbf{y}; \theta)$
- Q: What do the symbols mean?
  A: $\mathbf{X}$: data matrix; θ: learnable parameters; $\mathbf{y}$: ground truth vector;
- An example of loss/error/cost function: mean square error
  $L(\mathbf{X}, \mathbf{y}; \theta) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}^{(i)}; \theta) - y^{(i)})^2$
- Q: What does a partial derivative $\frac{\partial}{\partial \theta_k} L(\mathbf{X}; \mathbf{y}; \theta)$ represent?
  A: How a small change in a single parameter θ_k changes L.

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- Let the loss/error/cost function L be given by $L(\mathbf{X}; \mathbf{y}; \theta)$
- Q: What do the symbols mean?
  A: $\mathbf{X}$: data matrix; θ: learnable parameters; $\mathbf{y}$: ground truth vector;
- An example of loss/error/cost function: mean square error
  $L(\mathbf{X}, \mathbf{y}; \theta) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}^{(i)}; \theta) - y^{(i)})^2$
- Q: What does a partial derivative $\frac{\partial}{\partial \theta_k} L(\mathbf{X}; \mathbf{y}; \theta)$ represent?
  A: How a small change in a single parameter θ_k changes L.
- Q: What is the gradient $\nabla_\theta L(\mathbf{X}; \mathbf{y}; \theta)$?

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- Let the loss/error/cost function L be given by $L(\mathbf{X}; \mathbf{y}; \theta)$
- Q: What do the symbols mean?
  A: $\mathbf{X}$: data matrix; θ: learnable parameters; $\mathbf{y}$: ground truth vector;
- An example of loss/error/cost function: mean square error
  $L(\mathbf{X}, \mathbf{y}; \theta) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}^{(i)}; \theta) - y^{(i)})^2$
- Q: What does a partial derivative $\frac{\partial}{\partial \theta_k} L(\mathbf{X}; \mathbf{y}; \theta)$ represent?
  A: How a small change in a single parameter θ_k changes L.
- Q: What is the gradient $\nabla_\theta L(\mathbf{X}; \mathbf{y}; \theta)$?
  A: Vector of all partial derivatives (ie: derivatives of all parameters)

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- Let the loss/error/cost function L be given by $L(\mathbf{X}; \mathbf{y}; \theta)$
- Q: What do the symbols mean?
  A: $\mathbf{X}$: data matrix; θ: learnable parameters; $\mathbf{y}$: ground truth vector;
- An example of loss/error/cost function: mean square error
  $L(\mathbf{X}, \mathbf{y}; \theta) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}^{(i)}; \theta) - y^{(i)})^2$
- Q: What does a partial derivative $\frac{\partial}{\partial \theta_k} L(\mathbf{X}; \mathbf{y}; \theta)$ represent?
  A: How a small change in a single parameter θ_k changes L.
- Q: What is the gradient $\nabla_\theta L(\mathbf{X}; \mathbf{y}; \theta)$?
  A: Vector of all partial derivatives (ie: derivatives of all parameters)
- Q: What does this do: $\theta^* = \theta - \epsilon \nabla_\theta L(\mathbf{X}; \mathbf{y}; \theta)$?

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- Let the loss/error/cost function L be given by $L(\mathbf{X}; \mathbf{y}; \theta)$
- Q: What do the symbols mean?
  A: $\mathbf{X}$: data matrix; θ: learnable parameters; $\mathbf{y}$: ground truth vector;
- An example of loss/error/cost function: mean square error
  $L(\mathbf{X}, \mathbf{y}; \theta) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}^{(i)}; \theta) - y^{(i)})^2$
- Q: What does a partial derivative $\frac{\partial}{\partial \theta_k} L(\mathbf{X}; \mathbf{y}; \theta)$ represent?
  A: How a small change in a single parameter θ_k changes L.
- Q: What is the gradient $\nabla_\theta L(\mathbf{X}; \mathbf{y}; \theta)$?
  A: Vector of all partial derivatives (ie: derivatives of all parameters)
- Q: What does this do: $\theta^* = \theta - \epsilon \nabla_\theta L(\mathbf{X}; \mathbf{y}; \theta)$?
  A: Updates the parameters using gradient descent, where ϵ is the learning rate

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- The loss on all samples: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- The loss on all samples: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
- Gradient for all samples:

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- The loss on all samples: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
- Gradient for all samples: $\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m}$

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- The loss on all samples: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
- Gradient for all samples: $\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- The loss on all samples: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
- Gradient for all samples: $\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
  Q: Can you think of a reason why not use all samples?

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- The loss on all samples: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
- Gradient for all samples: $\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
  Q: Can you think of a reason why not use all samples?
  A: Huge datasets are impractical; takes too long to take a step.

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- The loss on all samples: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
- Gradient for all samples: $\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
  Q: Can you think of a reason why not use all samples?
  A: Huge datasets are impractical; takes too long to take a step.
- Stochastic Gradient Descent (SGD) computes exact gradients from a small number of samples k to approximate the gradient from all samples.

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- The loss on all samples: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
- Gradient for all samples: $\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
  Q: Can you think of a reason why not use all samples?
  A: Huge datasets are impractical; takes too long to take a step.
- Stochastic Gradient Descent (SGD) computes exact gradients from a small number of samples k to approximate the gradient from all samples.
- Q: How does SGD update θ^*?

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- The loss on all samples: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
- Gradient for all samples: $\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
  Q: Can you think of a reason why not use all samples?
  A: Huge datasets are impractical; takes too long to take a step.
- Stochastic Gradient Descent (SGD) computes exact gradients from a small number of samples k to approximate the gradient from all samples.
- Q: How does SGD update θ^*?
  A: $\theta^* = \theta - \epsilon \frac{1}{k} \sum_{i=1}^{k} \nabla_\theta L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$

Stochastic Gradient Descent

DL-book: 5.9; UDL-book: 6.2

- The loss on all samples: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
- Gradient for all samples: $\nabla_\theta J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$
  Q: Can you think of a reason why not use all samples?
  A: Huge datasets are impractical; takes too long to take a step.
- Stochastic Gradient Descent (SGD) computes exact gradients from a small number of samples k to approximate the gradient from all samples.
- Q: How does SGD update θ^*?
  A: $\theta^* = \theta - \epsilon \frac{1}{k} \sum_{i=1}^{k} \nabla_\theta L(\mathbf{x}^{(i)}, y^{(i)}, \theta)$

This is the procedure we've already seen.

Training a network

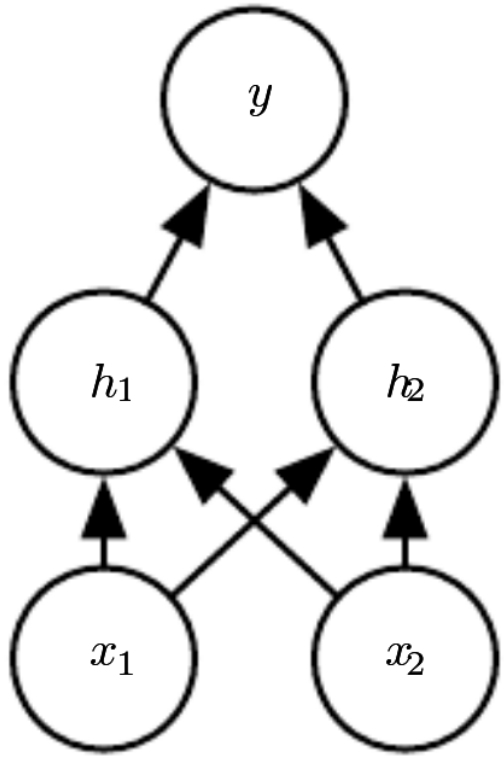
DL-book: 4.3

Training set:

Data		Label
0	1	1
1	1	0
0	0	0
1	0	1

While not converged:

1. Present a training sample
2. Compare the results
3. Update the parameters



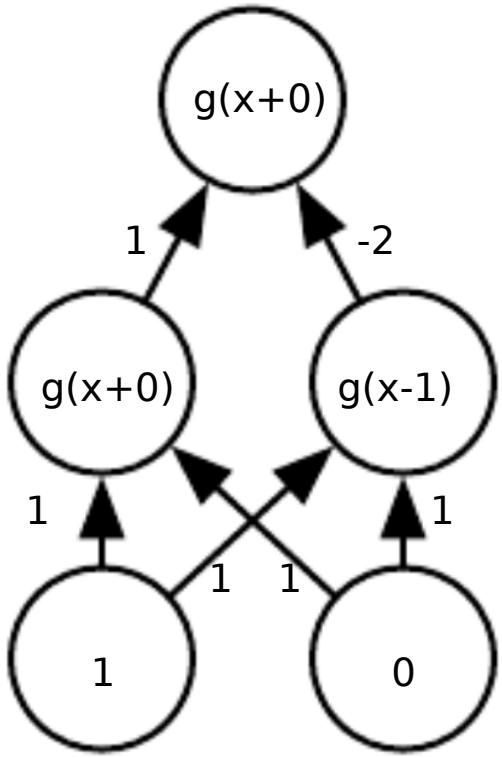
Questions?

Deep Feed forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1; UDL-book: 4

Slide 3: Network $f = g(\mathbf{x}^\top W + \mathbf{c}^\top)\mathbf{w} + b$

- Q: What is the equation of the last layer?

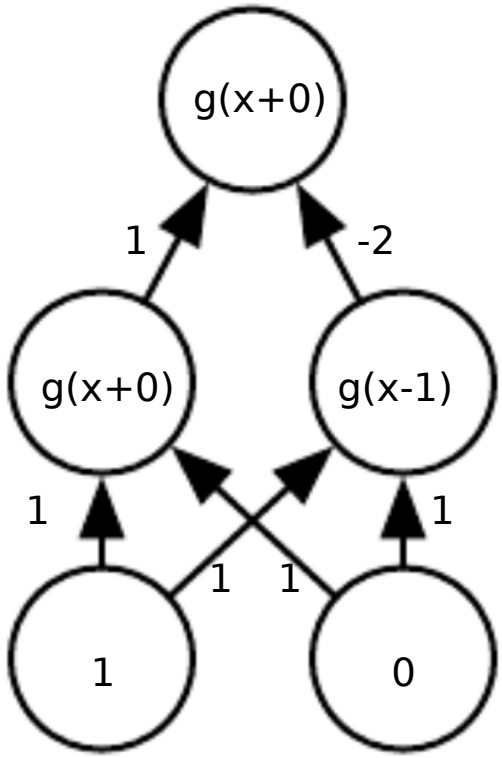


Deep Feed forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1; UDL-book: 4

Slide 3: Network $f = g(\mathbf{x}^\top W + \mathbf{c}^\top)\mathbf{w} + b$

- Q: What is the equation of the last layer?
  A: Let $\mathbf{h} = g(\mathbf{x}^\top W + \mathbf{c}^\top)^\top$; last layer: $\mathbf{h}^\top \mathbf{w} + b$

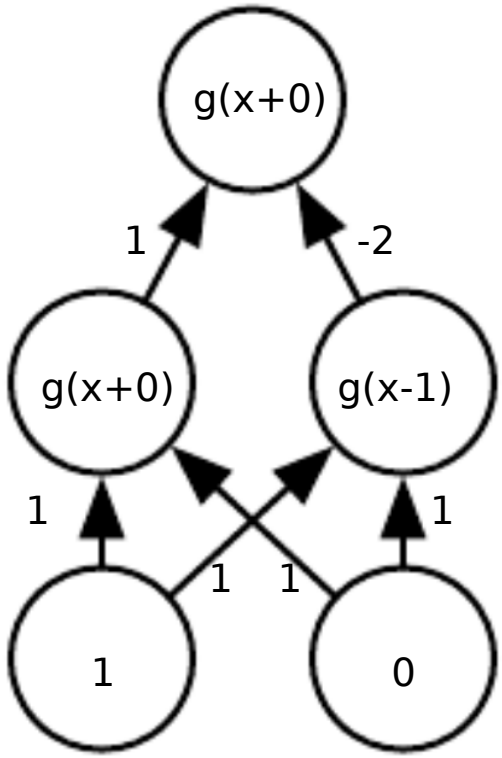


Deep Feed forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1; UDL-book: 4

Slide 3: Network $f = g(\mathbf{x}^\top W + \mathbf{c}^\top)\mathbf{w} + b$

- Q: What is the equation of the last layer?
  A: Let $\mathbf{h} = g(\mathbf{x}^\top W + \mathbf{c}^\top)^\top$; last layer: $\mathbf{h}^\top \mathbf{w} + b$
- Linear is good, but possibly too rigid (it can only do a line/plane)



Deep Feed forward Networks (Multi-layered perceptron)

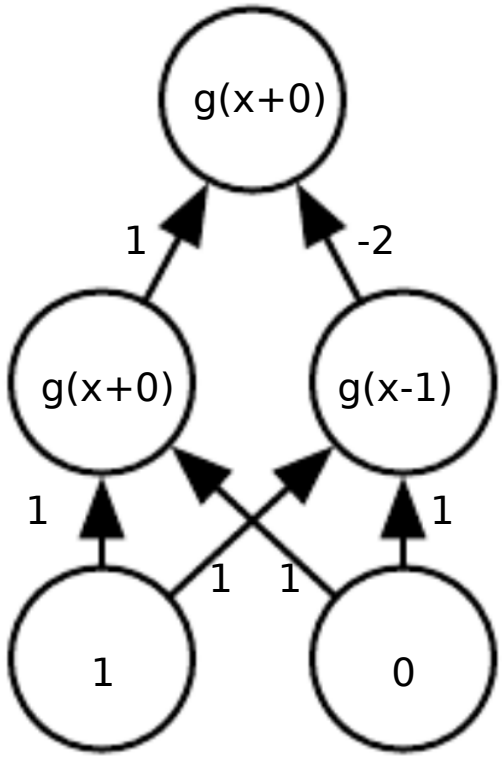
DL-book: 6, 6.1; UDL-book: 4

Slide 3: Network $f = g(\mathbf{x}^\top W + \mathbf{c}^\top)\mathbf{w} + b$

- Q: What is the equation of the last layer?
  A: Let $\mathbf{h} = g(\mathbf{x}^\top W + \mathbf{c}^\top)^\top$; last layer: $\mathbf{h}^\top \mathbf{w} + b$
- Linear is good, but possibly too rigid (it can only do a line/plane)

Three ways to make the input non-linear:

1. $\phi()$ as generic kernel function (eg. RBF kernels)
2. $\phi()$ as feature extractors (eg. SIFT)
3. Learn $\phi()$..



Deep Feed forward Networks (Multi-layered perceptron)

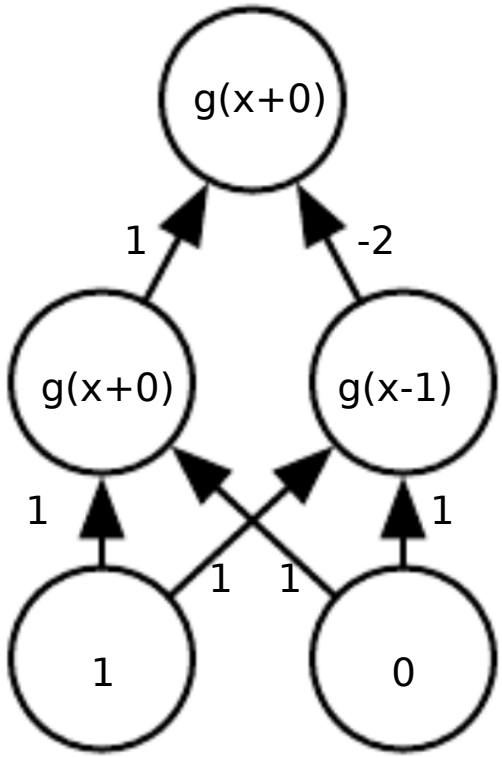
DL-book: 6, 6.1; UDL-book: 4

Slide 3: Network $f = g(\mathbf{x}^\top W + \mathbf{c}^\top)\mathbf{w} + b$

- Q: What is the equation of the last layer?
  A: Let $\mathbf{h} = g(\mathbf{x}^\top W + \mathbf{c}^\top)^\top$; last layer: $\mathbf{h}^\top \mathbf{w} + b$
- Linear is good, but possibly too rigid (it can only do a line/plane)

Three ways to make the input non-linear:

1. $\phi()$ as generic kernel function (eg. RBF kernels)
2. $\phi()$ as feature extractors (eg. SIFT)
3. Learn $\phi()$.. $y = f(\mathbf{x}, \theta, \mathbf{w}, b) = \phi(\mathbf{x}, \theta)^\top \mathbf{w} + b$

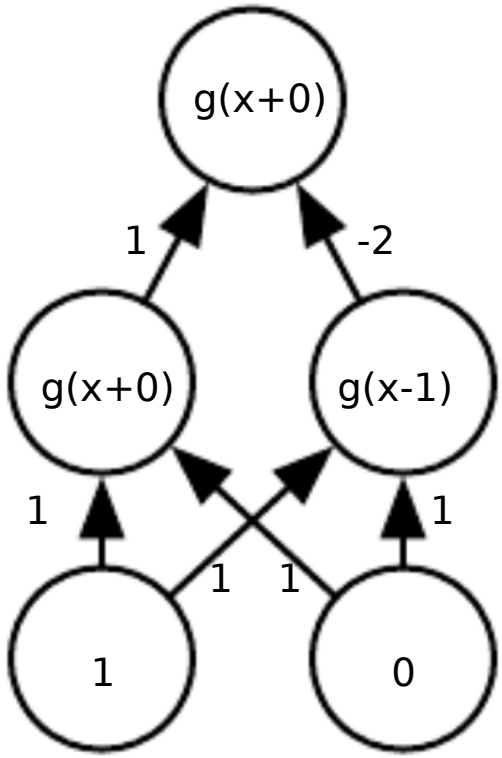


Deep Feed forward Networks (Multi-layered perceptron)

DL-book: 6, 6.1; UDL-book: 4

Slide 3: Network $f = g(\mathbf{x}^\top W + \mathbf{c}^\top)\mathbf{w} + b$

- Q: What is the equation of the last layer?
  A: Let $\mathbf{h} = g(\mathbf{x}^\top W + \mathbf{c}^\top)^\top$; last layer: $\mathbf{h}^\top \mathbf{w} + b$
- Linear is good, but possibly too rigid (it can only do a line/plane)



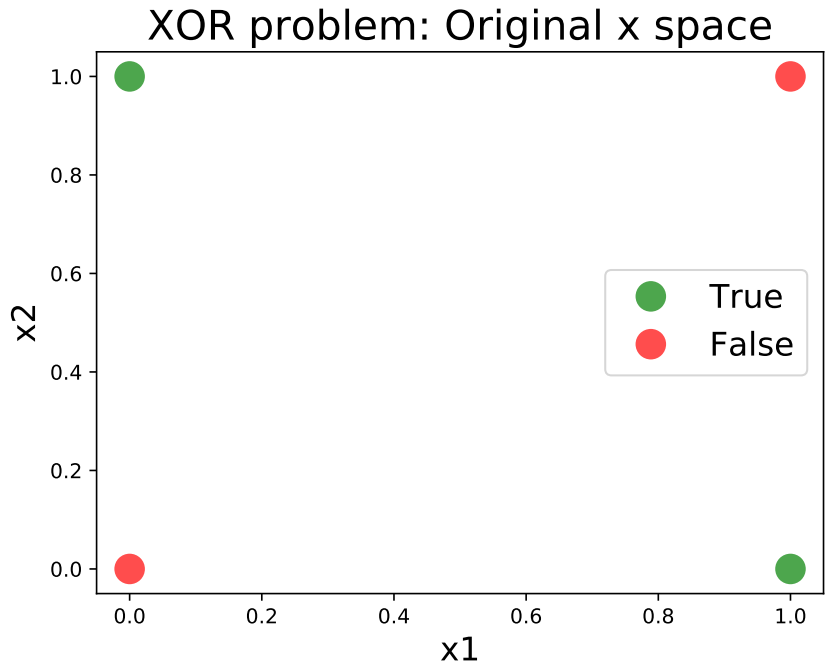
Three ways to make the input non-linear:

1. $\phi()$ as generic kernel function (eg. RBF kernels)
2. $\phi()$ as feature extractors (eg. SIFT)
3. Learn $\phi()$.. $\quad y = f(\mathbf{x}, \theta, \mathbf{w}, b) = \phi(\mathbf{x}, \theta)^\top \mathbf{w} + b$

(3) Deep learning: learns the representation (features)

Example: XOR

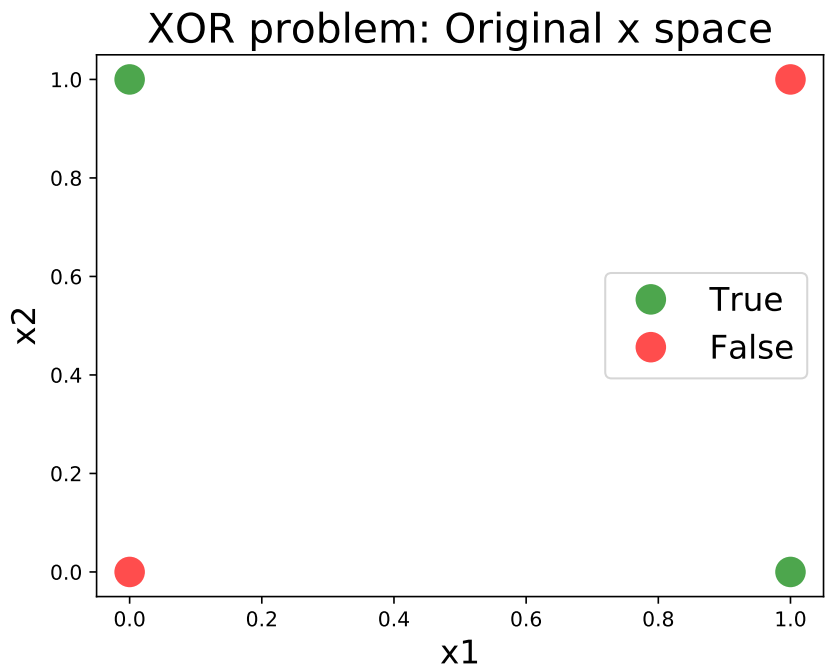
DL-book: 6, 6.1



Example: XOR

DL-book: 6, 6.1

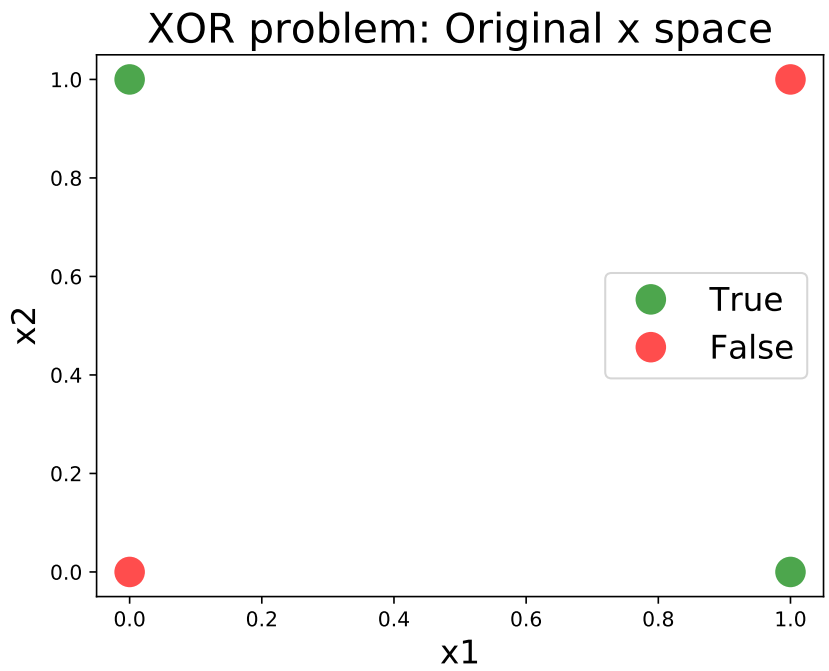
- Input: $\mathbb{X} = \{[0,0]^\top, [0,1]^\top, [1,0]^\top, [1,1]^\top\}$
- Labels: $\mathbb{Y} = \{0,1,1,0\}$



Example: XOR

DL-book: 6, 6.1

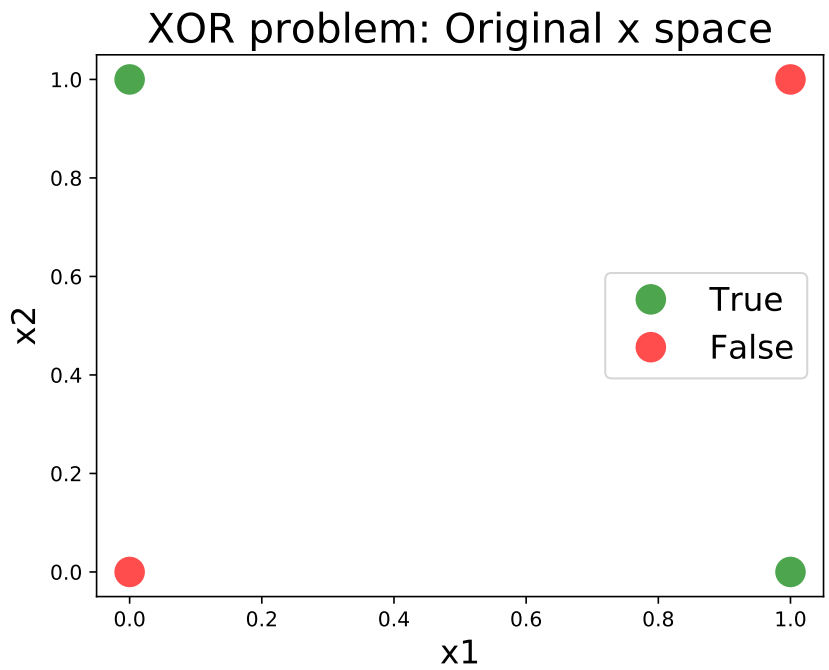
- Input: $\mathbb{X} = \{[0,0]^\top, [0,1]^\top, [1,0]^\top, [1,1]^\top\}$
- Labels: $\mathbb{Y} = \{0,1,1,0\}$
- Model: $f(\mathbf{x}, \mathbf{w}, b) = \mathbf{x}^\top \mathbf{w} + b$



Example: XOR

DL-book: 6, 6.1

- Input: $\mathbb{X} = \{[0,0]^\top, [0,1]^\top, [1,0]^\top, [1,1]^\top\}$
- Labels: $\mathbb{Y} = \{0,1,1,0\}$
- Model: $f(\mathbf{x}, \mathbf{w}, b) = \mathbf{x}^\top \mathbf{w} + b$

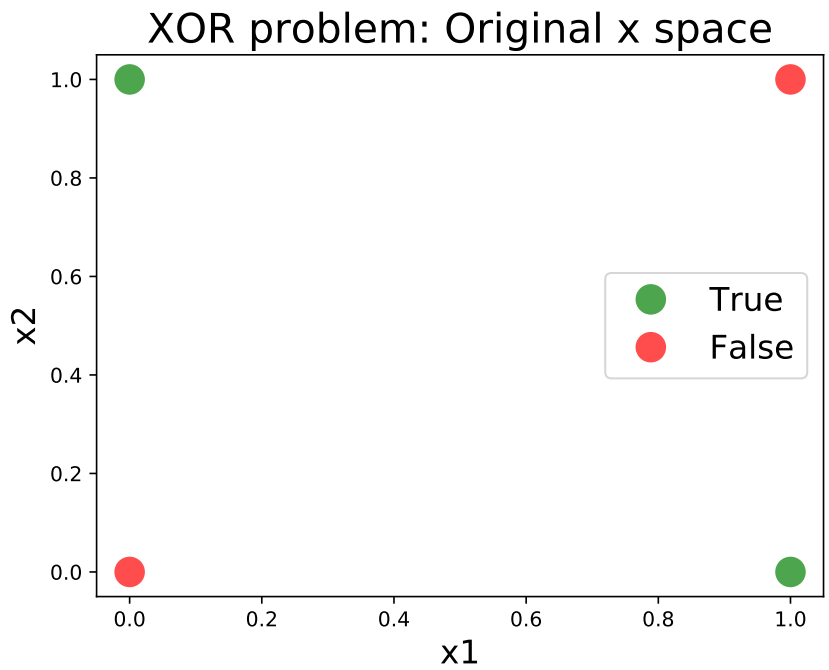


Q: Can it learn this?

Example: XOR

DL-book: 6, 6.1

- Input: $\mathbb{X} = \{[0,0]^\top, [0,1]^\top, [1,0]^\top, [1,1]^\top\}$
- Labels: $\mathbb{Y} = \{0,1,1,0\}$
- Model: $f(\mathbf{x}, \mathbf{w}, b) = \mathbf{x}^\top \mathbf{w} + b$

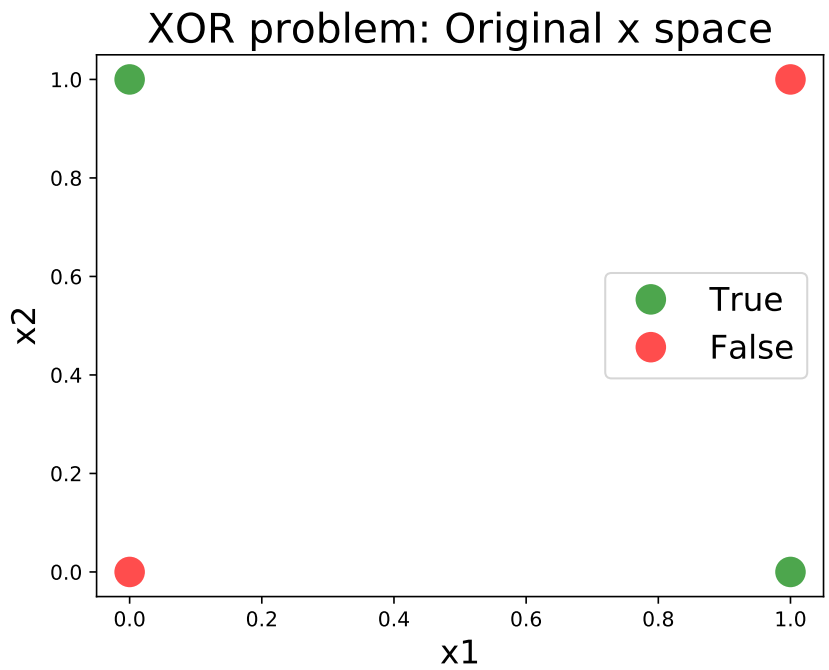


Q: Can it learn this? Q: Why not?

Example: XOR

DL-book: 6, 6.1

- Input: $\mathbb{X} = \{[0,0]^\top, [0,1]^\top, [1,0]^\top, [1,1]^\top\}$
- Labels: $\mathbb{Y} = \{0,1,1,0\}$
- Model: $f(\mathbf{x}, \mathbf{w}, b) = \mathbf{x}^\top \mathbf{w} + b$

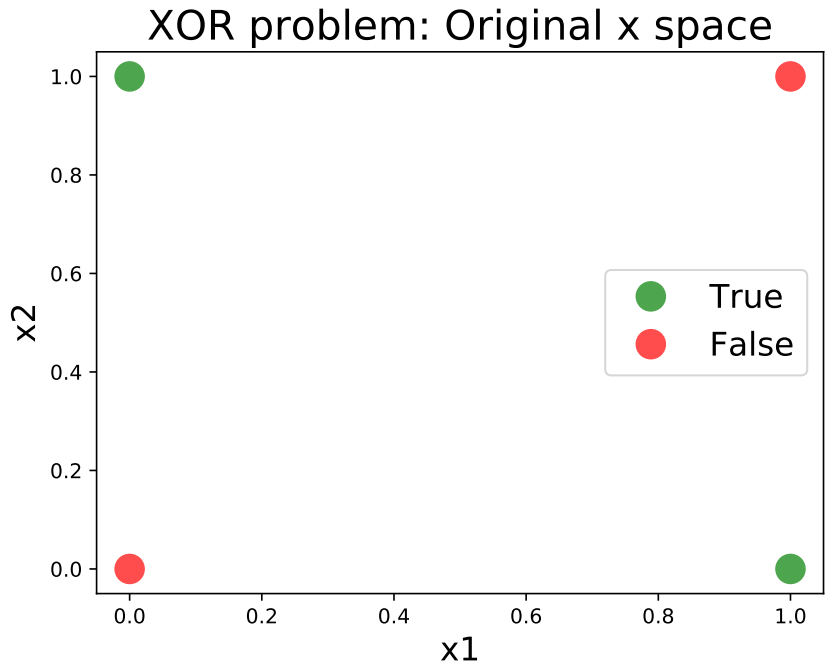


Q: Can it learn this? Q: Why not?
Example: $b = \frac{1}{2}, \mathbf{w} = [0,0]$; what does that do?

Example: XOR

DL-book: 6, 6.1

- Input: $\mathbb{X} = \{[0,0]^\top, [0,1]^\top, [1,0]^\top, [1,1]^\top\}$
- Labels: $\mathbb{Y} = \{0,1,1,0\}$
- Model: $f(\mathbf{x}, \mathbf{w}, b) = \mathbf{x}^\top \mathbf{w} + b$



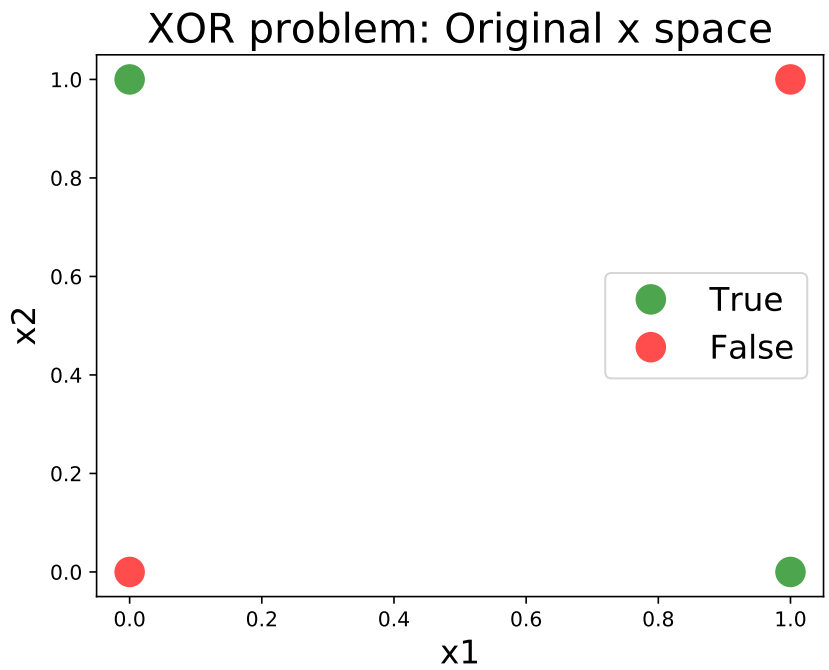
Q: Can it learn this? Q: Why not?

Example: $b = \frac{1}{2}, \mathbf{w} = [0,0]$; what does that do? Everything $\frac{1}{2}$

Example: XOR

DL-book: 6, 6.1

- Input: $\mathbb{X} = \{[0,0]^\top, [0,1]^\top, [1,0]^\top, [1,1]^\top\}$
- Labels: $\mathbb{Y} = \{0,1,1,0\}$
- Model: $f(\mathbf{x}, \mathbf{w}, b) = \mathbf{x}^\top \mathbf{w} + b$

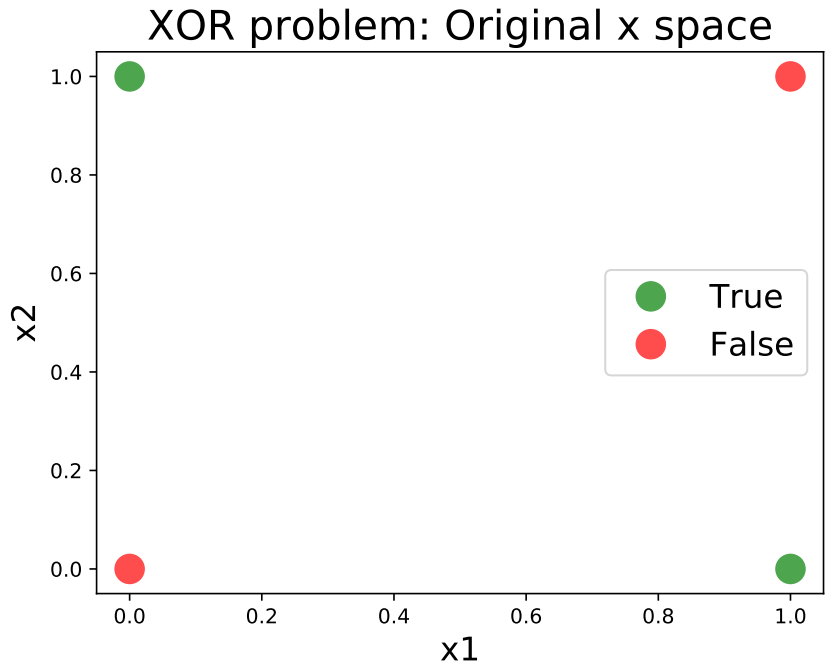


Q: Can it learn this? Q: Why not?
Example: $b = \frac{1}{2}, \mathbf{w} = [0,0]$; what does that do? Everything $\frac{1}{2}$
Q: Solution?

Example: XOR

DL-book: 6, 6.1

- Input: $\mathbb{X} = \{[0,0]^\top, [0,1]^\top, [1,0]^\top, [1,1]^\top\}$
- Labels: $\mathbb{Y} = \{0, 1, 1, 0\}$
- Model: $f(\mathbf{x}, \mathbf{w}, b) = \mathbf{x}^\top \mathbf{w} + b$

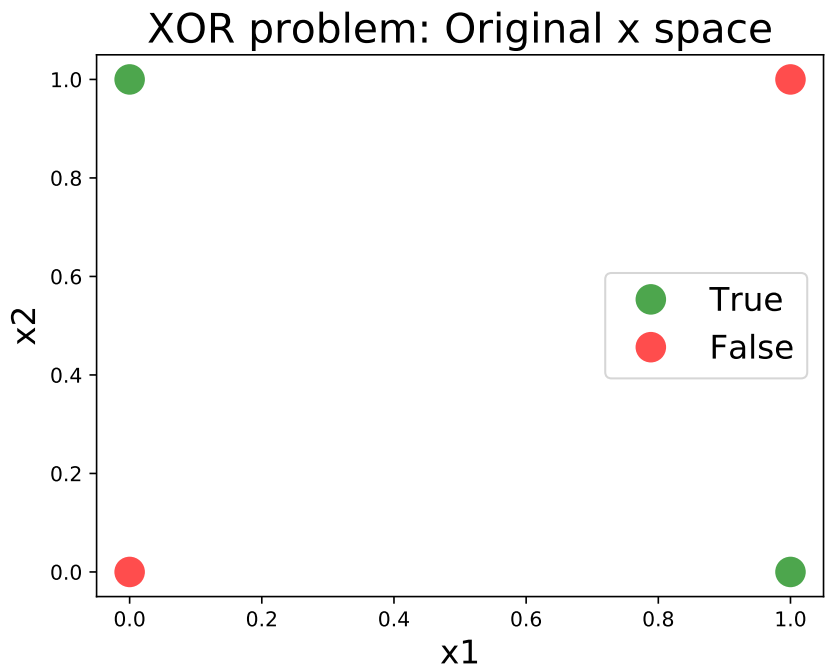


Q: Can it learn this? Q: Why not?
Example: $b = \frac{1}{2}, \mathbf{w} = [0, 0]$; what does that do? Everything $\frac{1}{2}$
Q: Solution? A: Add a hidden layer $\mathbf{h} = f^1(\mathbf{x}, \mathbf{W}, \mathbf{c})$ and $y = f^2(\mathbf{h}, \mathbf{w}, b)$

Example: XOR

DL-book: 6, 6.1

- Input: $\mathbb{X} = \{[0,0]^\top, [0,1]^\top, [1,0]^\top, [1,1]^\top\}$
- Labels: $\mathbb{Y} = \{0,1,1,0\}$
- Model: $f(\mathbf{x}, \mathbf{w}, b) = \mathbf{x}^\top \mathbf{w} + b$



Q: Can it learn this? Q: Why not?
Example: $b = \frac{1}{2}, \mathbf{w} = [0,0]$; what does that do? Everything $\frac{1}{2}$
Q: Solution? A: Add a hidden layer $\mathbf{h} = f^1(\mathbf{x}, \mathbf{W}, \mathbf{c})$ and $y = f^2(\mathbf{h}, \mathbf{w}, b)$

Complete model: $f(\mathbf{x}, W, \mathbf{c}, \mathbf{w}, b) = f^2(f^1(\mathbf{x}, \mathbf{W}, \mathbf{c}), \mathbf{w}, b) = f^2(f^1(\mathbf{x}))$

Activation function

DL-book: 6, 6.1; UDL-book: fig 3.1;

Complete model: $f(\mathbf{x}, W, \mathbf{c}, \mathbf{w}, b) = f^2(f^1(\mathbf{x}, \mathbf{W}, \mathbf{c}), \mathbf{w}, b) = f^2(f^1(\mathbf{x}))$

Activation function

DL-book: 6, 6.1; UDL-book: fig 3.1;

Complete model: $f(\mathbf{x}, W, \mathbf{c}, \mathbf{w}, b) = f^2(f^1(\mathbf{x}, \mathbf{W}, \mathbf{c}), \mathbf{w}, b) = f^2(f^1(\mathbf{x}))$

- What function should f^1 compute? Linear?

Activation function

DL-book: 6, 6.1; UDL-book: fig 3.1;

Complete model: $f(\mathbf{x}, W, \mathbf{c}, \mathbf{w}, b) = f^2(f^1(\mathbf{x}, \mathbf{W}, \mathbf{c}), \mathbf{w}, b) = f^2(f^1(\mathbf{x}))$

- What function should f^1 compute? Linear? no.
  $f^1(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$ and $f^2(\mathbf{h}) = \mathbf{w}^\top \mathbf{h}$ then $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{W}^\top \mathbf{x}$,

Activation function

DL-book: 6, 6.1; UDL-book: fig 3.1;

Complete model: $f(\mathbf{x}, W, \mathbf{c}, \mathbf{w}, b) = f^2(f^1(\mathbf{x}, \mathbf{W}, \mathbf{c}), \mathbf{w}, b) = f^2(f^1(\mathbf{x}))$

- What function should f^1 compute? Linear? no.
  $f^1(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$ and $f^2(\mathbf{h}) = \mathbf{w}^\top \mathbf{h}$ then $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{W}^\top \mathbf{x}$,
- Make non-linear by 'activation' function. $h = g(\mathbf{W}^\top \mathbf{x} + \mathbf{c})$

Activation function

DL-book: 6, 6.1; UDL-book: fig 3.1;

Complete model: $f(\mathbf{x}, W, \mathbf{c}, \mathbf{w}, b) = f^2(f^1(\mathbf{x}, \mathbf{W}, \mathbf{c}), \mathbf{w}, b) = f^2(f^1(\mathbf{x}))$

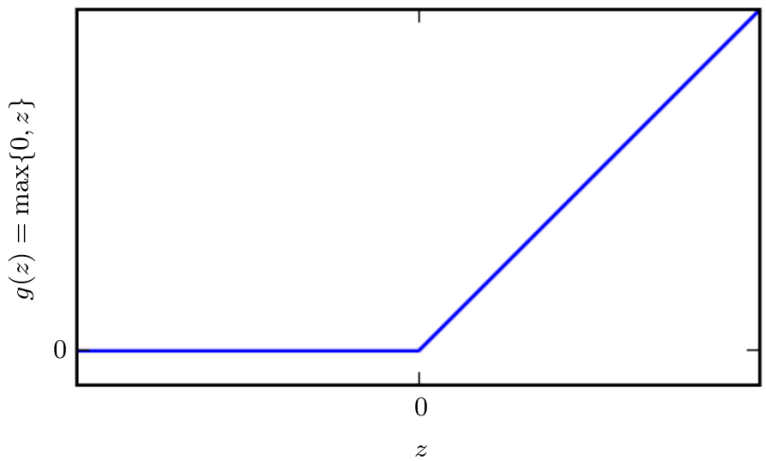
- What function should f^1 compute? Linear? no.
  $f^1(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$ and $f^2(\mathbf{h}) = \mathbf{w}^\top \mathbf{h}$ then $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{W}^\top \mathbf{x}$,
- Make non-linear by 'activation' function. $h = g(\mathbf{W}^\top \mathbf{x} + \mathbf{c})$
- Activation function applied **element-wise** for each i in h_i
- Rectified linear unit $g(z) = \max\{0, z\}$ (Q: graph?)

Activation function

DL-book: 6, 6.1; UDL-book: fig 3.1;

Complete model: $f(\mathbf{x}, W, \mathbf{c}, \mathbf{w}, b) = f^2(f^1(\mathbf{x}, \mathbf{W}, \mathbf{c}), \mathbf{w}, b) = f^2(f^1(\mathbf{x}))$

- What function should f^1 compute? Linear? no.
  $f^1(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$ and $f^2(\mathbf{h}) = \mathbf{w}^\top \mathbf{h}$ then $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{W}^\top \mathbf{x}$,
- Make non-linear by 'activation' function. $h = g(\mathbf{W}^\top \mathbf{x} + \mathbf{c})$
- Activation function applied **element-wise** for each i in h_i
- Rectified linear unit $g(z) = \max\{0, z\}$ (Q: graph?) A:



Solution to XOR

DL-book: 6, 6.1

Full model $f(\mathbf{X}, W, \mathbf{c}, \mathbf{w}, b) = \mathbf{w}^\top \max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} + b$

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad b = 0$$

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Q: can you validate this is a solution to

$$\mathbb{Y} = \{0, 1, 1, 0\} \text{ ?}$$

Solution to XOR

DL-book: 6, 6.1

Full model $f(\mathbf{X}, W, \mathbf{c}, \mathbf{w}, b) = \mathbf{w}^\top \max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} + b$

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad b = 0$$

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Q: can you validate this is a solution to

$$\mathbb{Y} = \{0, 1, 1, 0\} \text{ ?}$$

$$\mathbf{W}^\top \mathbf{X} =$$

Solution to XOR

DL-book: 6, 6.1

Full model $f(\mathbf{X}, W, \mathbf{c}, \mathbf{w}, b) = \mathbf{w}^\top \max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} + b$

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad b = 0$$

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Q: can you validate this is a solution to

$$\mathbb{Y} = \{0, 1, 1, 0\} \text{ ?}$$

$$\mathbf{W}^\top \mathbf{X} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

Solution to XOR

DL-book: 6, 6.1

Full model $f(\mathbf{X}, W, \mathbf{c}, \mathbf{w}, b) = \mathbf{w}^\top \max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} + b$

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad b = 0$$

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Q: can you validate this is a solution to

$$\mathbb{Y} = \{0, 1, 1, 0\} \text{ ?}$$

$$\mathbf{W}^\top \mathbf{X} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

$$\max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} =$$

Solution to XOR

DL-book: 6, 6.1

Full model $f(\mathbf{X}, W, \mathbf{c}, \mathbf{w}, b) = \mathbf{w}^\top \max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} + b$

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad b = 0$$

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Q: can you validate this is a solution to $\mathbb{Y} = \{0, 1, 1, 0\}$?

$$\mathbf{W}^\top \mathbf{X} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

$$\max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution to XOR

DL-book: 6, 6.1

Full model $f(\mathbf{X}, W, \mathbf{c}, \mathbf{w}, b) = \mathbf{w}^\top \max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} + b$

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad b = 0$$

Q: plot $\max\{0, W^\top X + \mathbf{c}\}$ in 2d?

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Q: can you validate this is a solution to $\mathbb{Y} = \{0, 1, 1, 0\}$?

$$\mathbf{W}^\top \mathbf{X} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

$$\max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution to XOR

DL-book: 6, 6.1

Full model $f(\mathbf{X}, W, \mathbf{c}, \mathbf{w}, b) = \mathbf{w}^\top \max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} + b$

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad b = 0$$

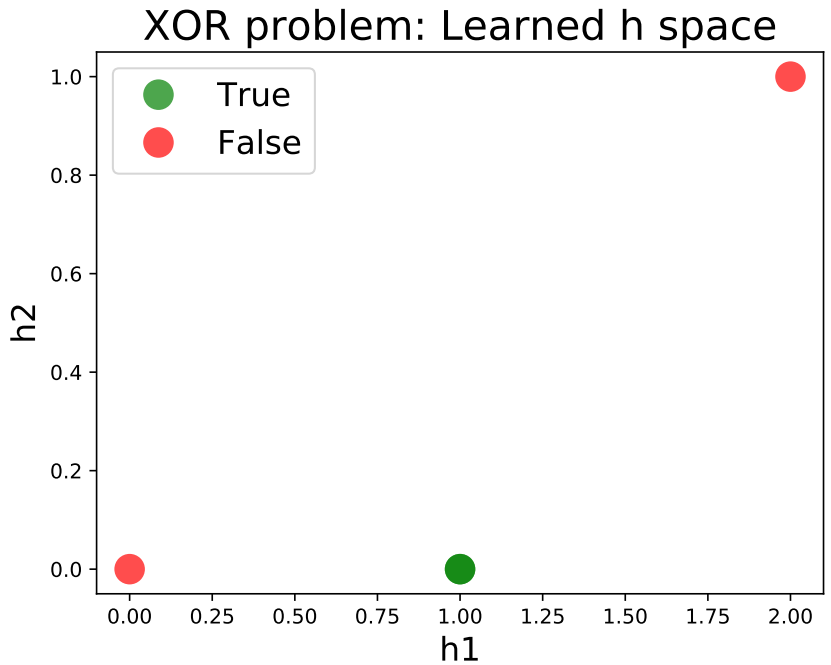
$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Q: can you validate this is a solution to $\mathbb{Y} = \{0, 1, 1, 0\}$?

$$\mathbf{W}^\top \mathbf{X} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

$$\max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Q: plot $\max\{0, W^\top X + \mathbf{c}\}$ in 2d?



Solution to XOR

DL-book: 6, 6.1

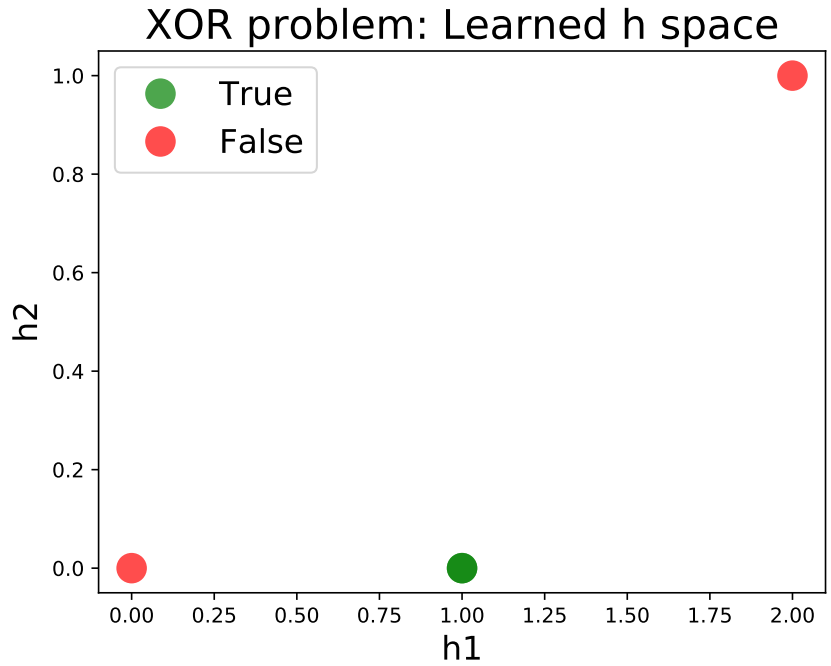
Full model $f(\mathbf{X}, W, \mathbf{c}, \mathbf{w}, b) = \mathbf{w}^\top \max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} + b$

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad b = 0$$

Q: plot $\max\{0, W^\top X + \mathbf{c}\}$ in 2d?

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Q: can you validate this is a solution to $\mathbb{Y} = \{0, 1, 1, 0\}$?



$$\mathbf{W}^\top \mathbf{X} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

$$\max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: $\mathbf{w}^\top \max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} + b =$

Solution to XOR

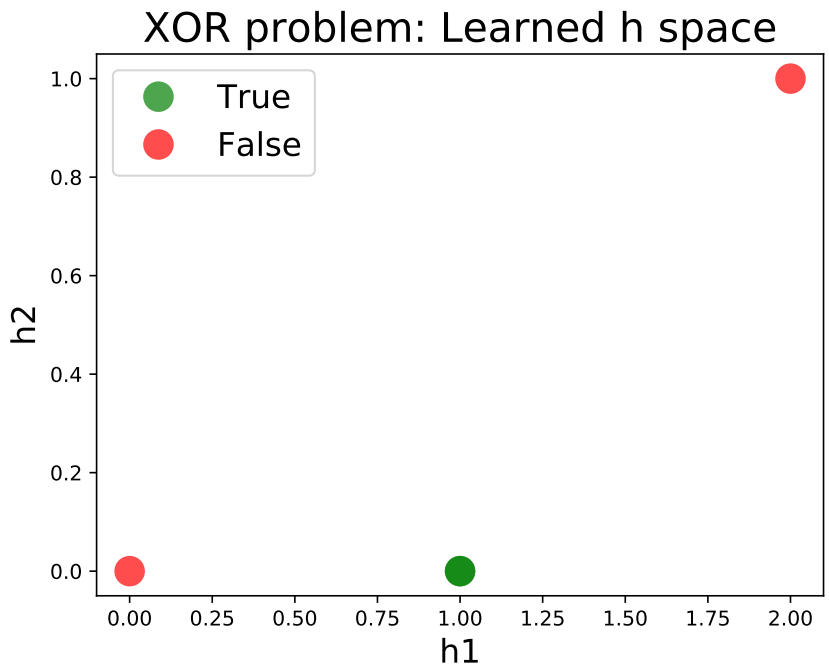
DL-book: 6, 6.1

Full model $f(\mathbf{X}, W, \mathbf{c}, \mathbf{w}, b) = \mathbf{w}^\top \max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} + b$

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad b = 0$$

Q: plot $\max\{0, W^\top X + \mathbf{c}\}$ in 2d?

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$



Q: can you validate this is a solution to $\mathbb{Y} = \{0, 1, 1, 0\}$?

$$\mathbf{W}^\top \mathbf{X} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

$$\max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: $\mathbf{w}^\top \max\{0, \mathbf{W}^\top \mathbf{X} + \mathbf{c}\} + b = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}$

Questions?

Summary

- Feed forward networks
- How to train them using SGD
- How representations are learned (XOR example)