Perceptrons

Young Wu
Based on lecture slides by Jerry Zhu

February 5, 2019

Recall From Previous Lecture

Supervised Learning

- Supervised learning:

Data	Features (Input)	Output	-
Sample	$\{(x_{i1}, ..., x_{iD})\}_{i=1}^{N}$	$\{y_i\}_{i=1}^{N}$	find "best" $\hat{f}$
-	observable	known	-
New	$\{(x'_1, ..., x'_D)\}_{i=1}^{N}$	y'	guess $\hat{y} = \hat{f}(x')$
-	observable	unknown	-

Recall From Previous Lecture

Training and Test Sets

- Supervised learning:

Data	Features (Input)	Output	-
Train	$\{(x_{i1}, \ldots, x_{iD})\}_{i=1}^{N_T}$	$\{y_i\}_{i=1}^{N}$	find many $\hat{f}$
-	observable	known	-
Test	$\{(x_{i1}, \ldots, x_{iD})\}_{i=1}^{N-N_T}$	$\{y_i\}_{i=1}^{N}$	find "best" $\hat{f}$
-	observable	known	-
New	$\{(x'_1, \ldots, x'_D)\}_{i=1}^{N}$	y'	guess $\hat{y} = \hat{f}(x')$
-	observable	unknown	-

Recall From Previous Lecture

Real Life Examples

- Examples of supervised learning:

Data	Features (Input)	Output
Images	x_{i1} = How much red?	$y_i \in \{0, 1\}$
-	x_{i2} = How many circles?	$0 =$ cat,
-	...	$1 =$ dog
Sentences	x_{i1} = Length?	$y_i \in \{0, 1\}$
-	x_{i2} = How many verbs?	$0 =$ bad news
-	...	$1 =$ good news
Medical	x_{i1} = Age	$y_i \in \{0, 1\}$
-	x_{i2} = Height	$0 =$ not sick
-	...	$1 =$ sick
Grades	x_{i1} = Midterm?	$y_i \in \{0, 1\}$
-	x_{i2} = Average homework?	$0 =$ fail
-	...	$1 =$ pass

Recall From Previous Lecture

Objective Function

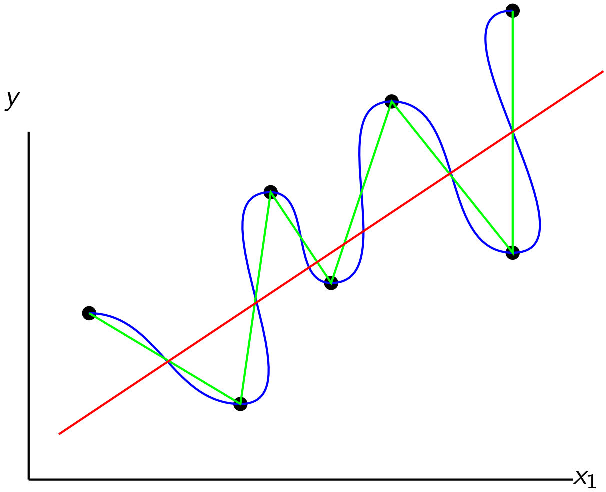
- How to select $\hat{f}$? Need an objective function

$$\hat{f} = \arg \min_{f \in \text{ all functions}} \frac{1}{2} \sum_{i=1}^{N} \left(f\left(x_i\right) - y_i\right)^2$$

- Problem: too many functions to choose from

Recall From Previous Lecture

Function Space Diagram



Recall From Previous Lecture

Hypothesis Space

- How to select $\hat{f}$? Need a hypothesis space

$$\hat{f} = \arg \min_{f \in \text{ all linear functions}} \frac{1}{2} \sum_{i=1}^{N} \left(f\left(x_i\right) - y_i\right)^2$$

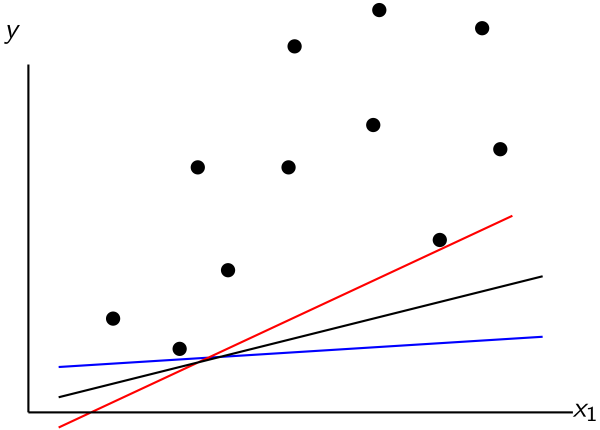
$$\left(\hat{w}_0, \hat{w}_1, ..., \hat{w}_D\right) = \arg \min_{(w_0, w_1, ..., w_D) \in \mathbb{R}^D} \frac{1}{2} \sum_{i=1}^{N} \left(a_i - y_i\right)^2$$

$$\text{where } a_i = w_0 + w_1 x_{i1} + w_2 x_{i1} + ... + w_D x_{iD}$$

- Problem: need an algorithm to solve the minimization problem

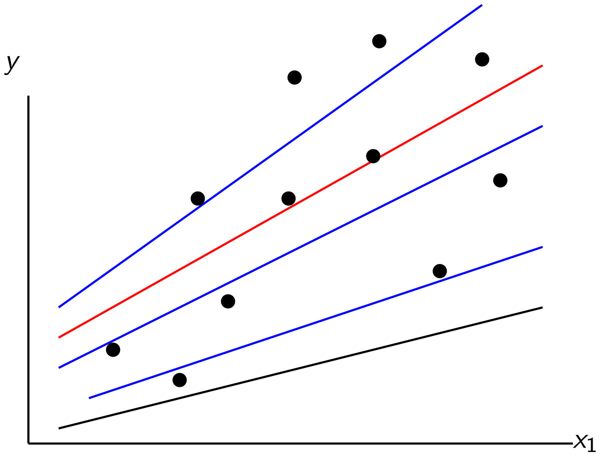
Recall From Previous Lecture

Optimization Diagram



Recall From Previous Lecture

Optimization Diagram, Converge



Recall From Previous Lecture

Optimization Intuition

- If a small increase in w_d causes the distances from the points to the regression line to decrease: increase w_d
- If a small increase in w_d causes the distances from the points to the regression line to increase: decrease w_d
- Change in distance due to change in w_d is the derivative
- Change in distance due to change in w is the gradient

Recall From Previous Lecture

Gradient Descent

- How to select $\hat{f}$? Need gradient descent

$$(\hat{w}_0, \hat{w}_1, ..., \hat{w}_D) = \arg \min_{(w_0, w_1, ..., w_D) \in \mathbb{R}^D} \frac{1}{2} \sum_{i=1}^{N} (a_i - y_i)^2$$

$$\text{where } a_i = w_0 + w_1 x_{i1} + w_2 x_{i1} + ... + w_D x_{iD}$$

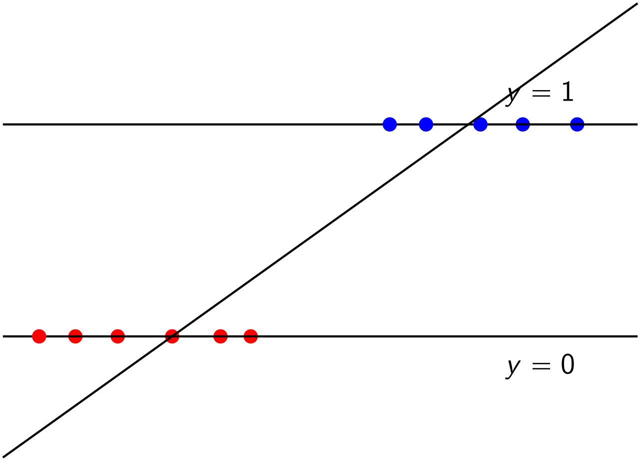
$$w_d = w_d - \alpha \sum_{i=1}^{N} (a_i - y_i) x_{id},$$

$$\text{for } d = 0, 1, ..., D, x_{i0} := 1$$

- Problem: not for binary classification

This Lecture

Binary Classification Diagram



This Lecture

Binary Classification Intuition

- The prediction $\hat{y}$ is not between 0 and 1
- Large x_i are classified correctly but have large distances from the regression line

This Lecture

Activation Function

- How to select $\hat{f}$ for binary classification? Need an activation function.

$$\hat{f} = \arg \min_{f = g(w_0 + w_1 x_{i1} + w_2 x_{i1} + ... + w_D x_{iD})} \frac{1}{2} \sum_{i=1}^{N} (a_i - y_i)^2$$

where $a_i = g\left(w_0 + w_1 x_{i1} + w_2 x_{i1} + ... + w_D x_{iD}\right)$

- Obvious choice: step function

Non-linear Activation Function

Step Function

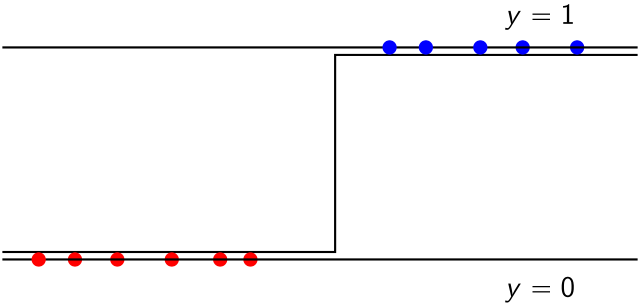
- Activation function: step function $g\left(\boxed{\cdot}\right) = \mathbb{1}_{\boxed{\cdot} \geqslant 0}$

$$\mathbb{1}_{\boxed{\cdot} \geqslant 0} = \begin{cases} 1 & \text{if } \boxed{\cdot} \geqslant 0 \\ 0 & \text{if } \boxed{\cdot} < 0 \end{cases}$$

- Derivative: $g'\left(\boxed{\cdot}\right) = 0$ but undefined at $\boxed{\cdot} = 0$
- Problem: discontinuous, cannot use gradient

Non-linear Activation Function

Step Function Diagram



Another Non-linear Activation Function

Sigmoid Function

- Activation function: sigmoid function $g\left(\boxed{\cdot}\right) = \dfrac{1}{1 + \exp\left(-\boxed{\cdot}\right)}$
- Derivative: $g'\left(\boxed{\cdot}\right) = g\left(\boxed{\cdot}\right)\left(1 - g\left(\boxed{\cdot}\right)\right)$
- Gradient descent step:

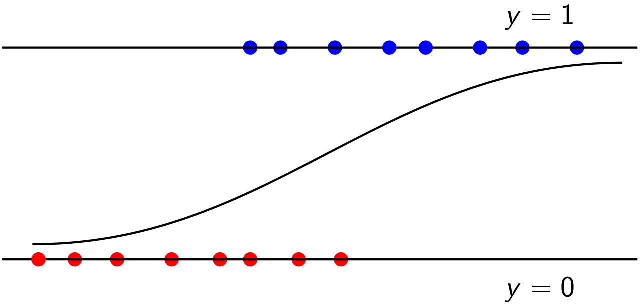
$$w_d = w_d - \alpha \sum_{i=1}^{N} \left(a_i - y_i\right) a_i \left(1 - a_i\right) x_{id}$$

where $a_i = g\left(w_0 + w_1 x_{i1} + w_2 x_{i1} + \ldots + w_D x_{iD}\right)$

for $d = 0, 1, \ldots, D, x_{i0} := 1$

Non-linear Activation Function

Sigmoid Function Diagram



Other Non-linear Activation Function

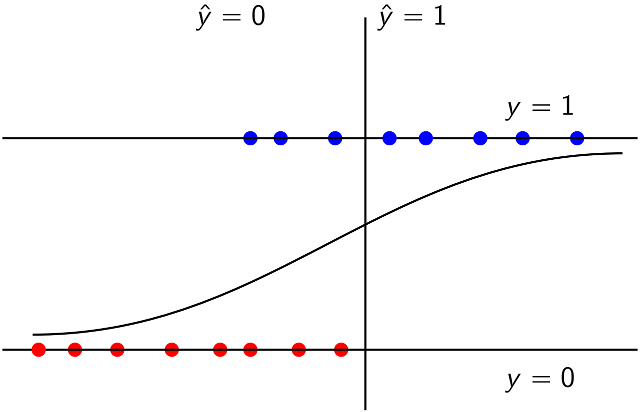
Tanh, Acrtan and ReLu

- Activation function: $g\left(\boxed{\cdot}\right) = \tanh\left(\boxed{\cdot}\right) = \dfrac{e^{\boxed{\cdot}} - e^{\boxed{\cdot}}}{e^{\boxed{\cdot}} + e^{\boxed{\cdot}}}$
- Activation function: $g\left(\boxed{\cdot}\right) = \arctan\left(\boxed{\cdot}\right)$
- Activation function (rectified linear unit): $g\left(\boxed{\cdot}\right) = \boxed{\cdot}\,\mathbb{1}_{\boxed{\cdot}\geqslant 0}$

- Gradient descent: all convex and differentiable.
- Problem: decision boundary is still step function

Other Non-linear Activation Function

Decision Boundary



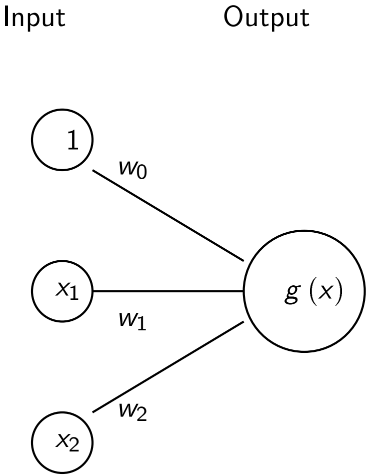
Multi-layer Neural Network

Intuition

- $f(x_i) = g(w_0 + w_1 x_{i1} + w_2 x_{i2} + \ldots_{w_D} x_{iD})$ is called a perceptron model
- Connect perceptrons into a network

Multi-layer Neural Network

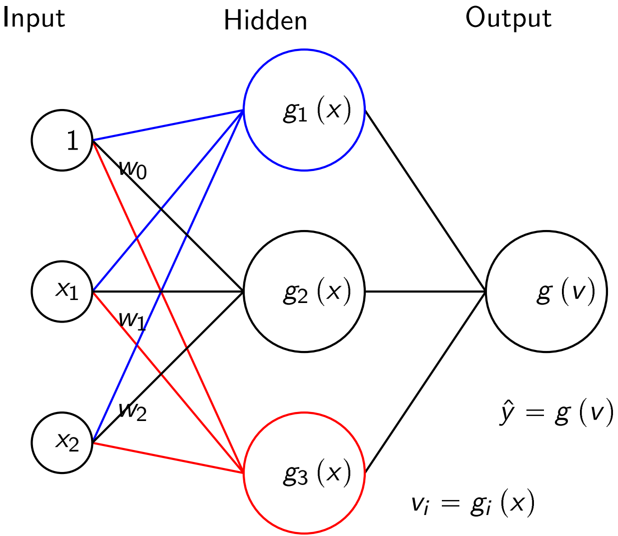
Single Layer Diagram



$$\hat{y} = g(x)$$

Multi-layer Neural Network

Multi Layer Diagram



Multi-layer Neural Network

Power

- In theory:

1. 1 Hidden-layer with enough hidden units can represent any continuous function of the inputs with arbitrary accuracy
2. 2 Hidden-layer can represent discontinuous functions

- In practice:

1. AlexNet: 8 layers
2. GoogLeNet: 27 layers
3. ResNet: 152 layers

Multi-layer Neural Network

Multi-class Classification

- Encode $y \in \{1, 2, 3, ..., K\}$ by using K output units

1. Class 1 $= (1, 0, 0, ..., 0, 0)$
2. Class 2 $= (0, 1, 0, ..., 0, 0)$
3. ...
4. Class $K = (0, 0, 0, ..., 0, 1)$

- Decode by choosing the class corresponding to the largest output unit

Next Lecture

Training Neural Network

- Derivatives are difficult to compute: use chain rule
- Backproprogation