CS540 Introduction to Artificial Intelligence
Lecture 4

Young Wu
Based on lecture slides by Jerry Zhu and Yingyu Liang

May 29, 2019

Test

Quiz (Graded)

- A:
- B:
- C:
- D: Choose this.
- E:

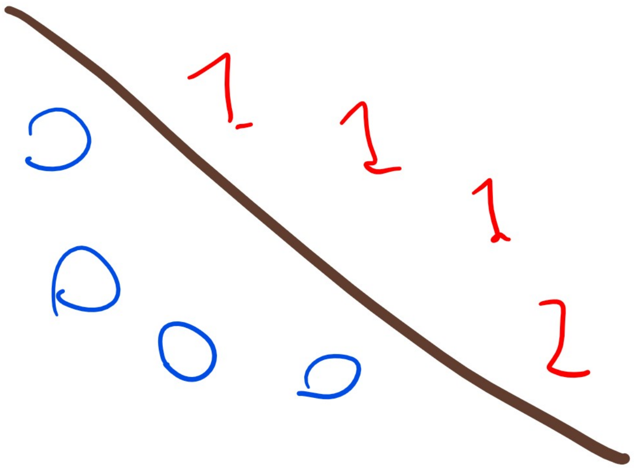
Homework

Quiz (Participation)

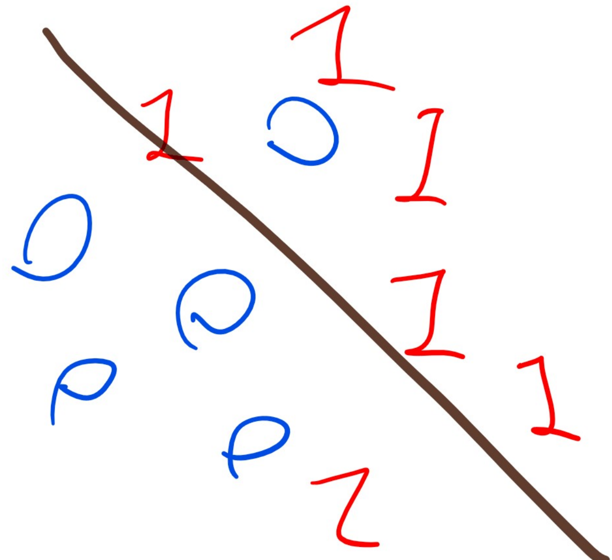
- Have you finished homework 1
- A: Waiting for solution.
- B: Will start soon.
- C: Started.
- D: Does not work due to bugs.
- E: Finished: 90+ percent accuracy.

Neural Network Diagram

Review



Perceptron Algorithm

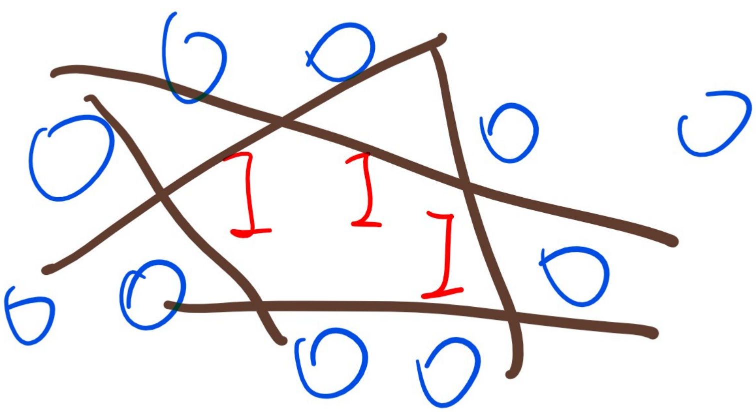


Logistic Regression

Gradient Descent

$$w = w - \alpha \nabla_w C$$

opposite direction



Neural Network

Multi-Layer Neural Network Diagram

Review

$$\nabla_w C = \sum_{i=1}^{n} \frac{\partial C}{\partial a_i} \cdot \nabla_w a_i$$

slow.

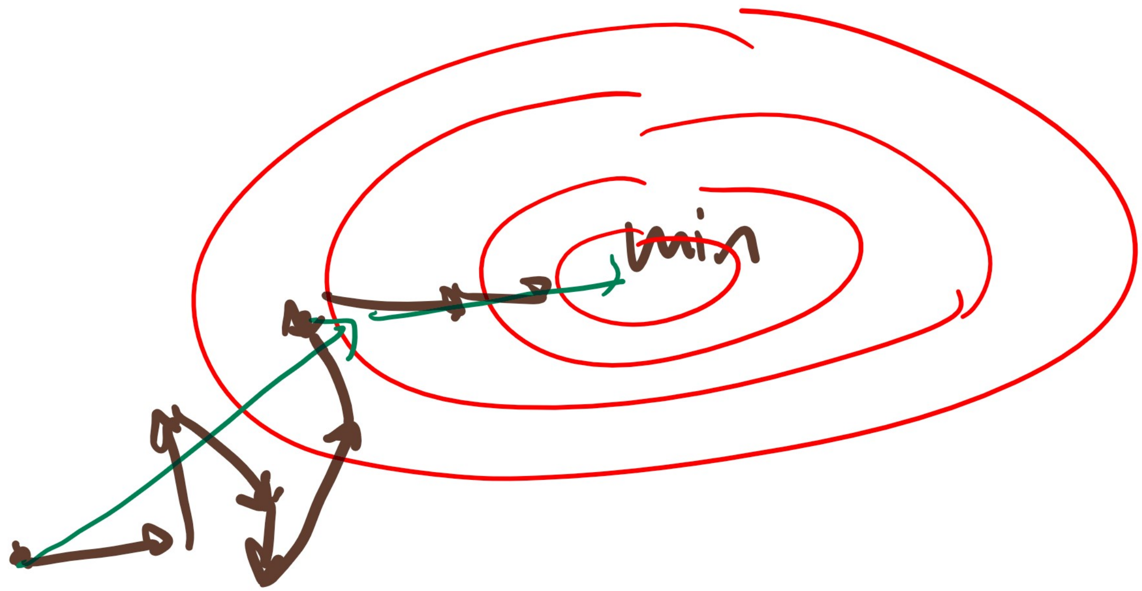
Stochastic Gradient Descent

Motivation

- Each gradient descent step requires the computation of gradients for all training instances $i = 1, 2, ..., n$. It is very costly.
- Stochastic gradient descent picks one instance x_i randomly, compute the gradient, and update the weights and biases.
- When a batch of instances is selected randomly each time, it is called batch gradient descent.

Stochastic Gradient Descent Diagram

Motivation



Stochastic Gradient Descent, Part 1

Algorithm

- Inputs, Outputs: same as backpropagation.
- Initialize the weights.
- Randomly permute (shuffle) the training set. Evaluate the activation functions at one instance at a time.
- Compute the gradient using the chain rule.

no sum over i

weight at layer l

$$\frac{\partial C}{\partial w_{j'j}^{(l)}} = \delta_{ij}^{(l)} a_{ij'}^{(l-1)}$$

$$\frac{\partial C}{\partial b_j^{(l)}} = \delta_{ij}^{(l)}$$

Stochastic Gradient Descent, Part 2

Algorithm

- Update the weights and biases using gradient descent.

For $l = 1, 2, ..., L$

$$w_{j'j}^{(l)} \leftarrow w_{j'j}^{(l)} - \alpha \frac{\partial C}{\partial w_{j'j}^{(l)}}, j' = 1, 2, ..., m^{(l-1)}, j = 1, 2, ..., m^{(l)}$$

$$b_{j}^{(l)} \leftarrow b_{j}^{(l)} - \alpha \frac{\partial C}{\partial b_{j}^{(l)}}, j = 1, 2, ..., m^{(l)}$$

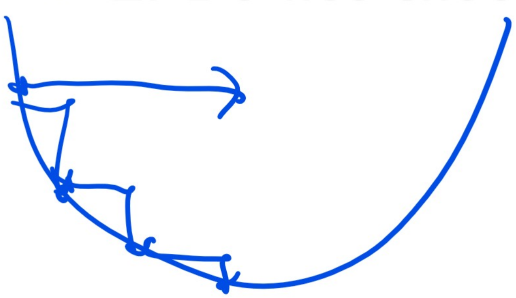
- Repeat the process until convergent.

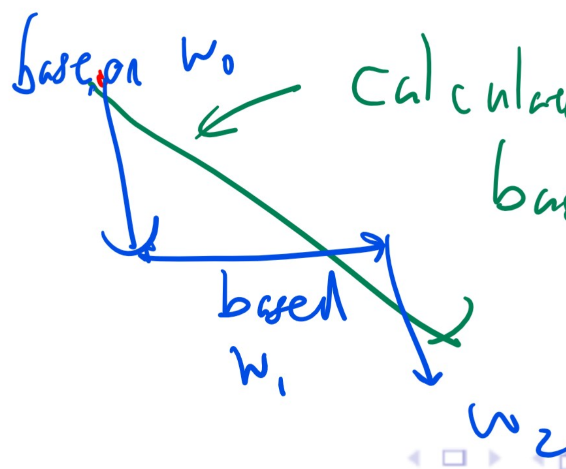
$$|C - C^{\text{prev}}| < \varepsilon$$

Stochastic vs Full Gradient Descent

Quiz (Participation)

- Given the same initial weights and biases, stochastic gradient descent with instances picked randomly without replacement and full gradient descent lead to the same updated weights.
- A: Do not choose this.
- B: True.
- C: Do not choose this.
- D: False.
- E: Do not choose this.





Calculate all $\underline{n}$ gradient based on initial w.

Generalization Error

Motivation

- With a large number of hidden units and small enough learning rate α, a multi-layer neural network can fit every finite training set perfectly.
- It does not imply the performance on the test set will be good.
- This problem is called overfitting.

Generalization Error Diagram

Motivation

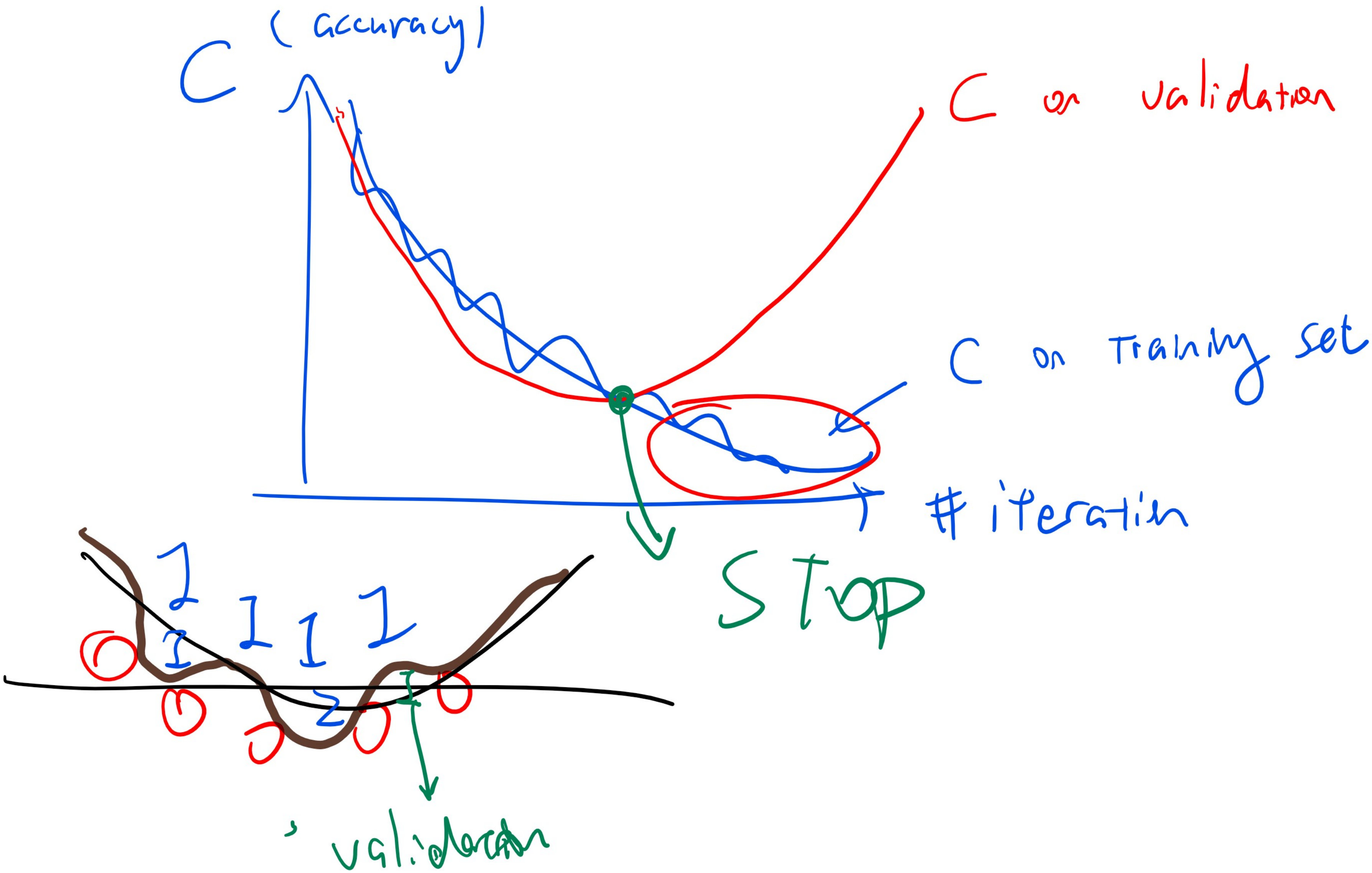
Method 1, Validation Set

Discussion

- Set aside a subset of the training set as the validation set.
- During training, the cost (or accuracy) on the training set will always be decreasing until it hits 0.
- Train the network until the cost (or accuracy) on the validation set begins to increase.

Validation Set Diagram

Discussion

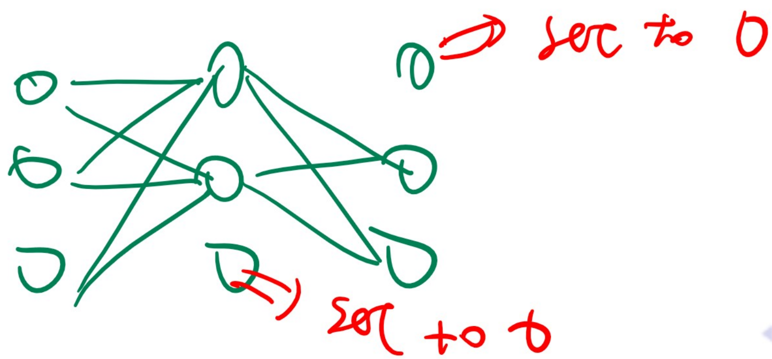


Method 2, Drop Out

Discussion

only NN

- At each hidden layer, a random set of units from that layer is set to 0.
- For example, each unit is retained with probability $p = 0.5$. During the test, the activations are reduced by $p = 0.5$ (or 50 percent).
- The intuition is that if a hidden unit works well with different combinations of other units, it does not rely on other units and it is likely to be individually useful.



Drop Out Diagram

Discussion

Method 3, L1 and L2 Regularization

Discussion

- The idea is to include an additional cost for non-zero weights.
- The models are simpler if many weights are zero.
- For example, if logistic regression has only a few non-zero weights, it means only a few features are relevant, so only these features are used for prediction.

Method 3, L1 Regularization

Discussion

- For L1 regularization, add the 1-norm of the weights to the cost.

$$C = \sum_{i=1}^{n} (a_i - y_i)^2 + \lambda \left\| \begin{bmatrix} w \\ b \end{bmatrix} \right\|_1 = \sum_{i=1}^{n} (a_i - y_i)^2 + \lambda \left(\sum_{i=1}^{m} |w_i| + |b| \right)$$

cost of having non-zero weights.

min C $\Rightarrow$ try to make $w \approx 0$

- Linear regression with L1 regularization is called LASSO (least absolute shrinkage and selection operator).

Method 3, L2 Regularization

Discussion

- For L2 regularization, add the 2-norm of the weights to the cost.

$$C = \sum_{i=1}^{n} (a_i - y_i)^2 + \frac{\lambda}{2} \left\| \begin{bmatrix} w \\ b \end{bmatrix} \right\|_2^2$$

$$= \sum_{i=1}^{n} (a_i - y_i)^2 + \underbrace{\frac{\lambda}{2} \left(\sum_{i=1}^{m} w_i^2 + b^2 \right)}$$

learning rate

$$w = w - \alpha \nabla_w C - \lambda w$$

regularization parameter.

easy for gradient descent.

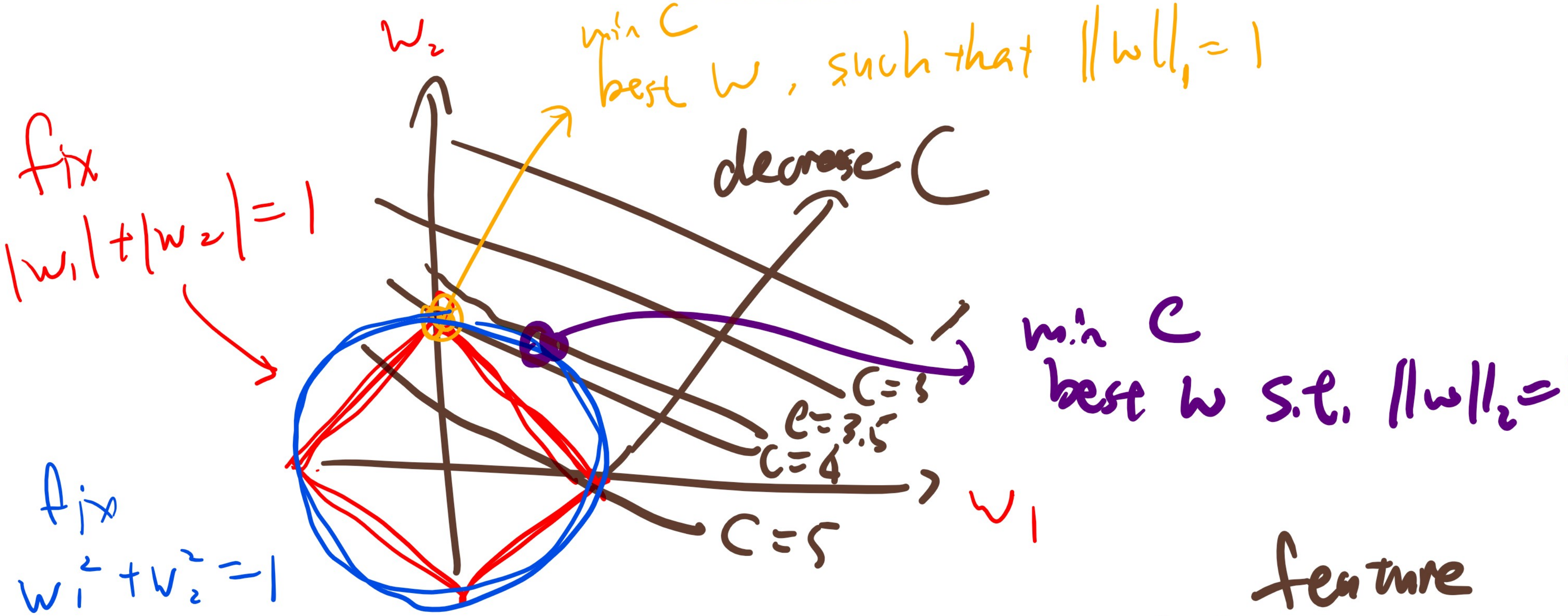
L1 and L2 Regularization Comparison

Discussion

- L1 regularization leads to more weights that are exactly 0. It is useful for feature selection.
- L2 regularization leads to more weights that are close to 0. It is easier to do gradient descent because 1-norm is not differentiable.

L1 and L2 Regularization Diagram

Discussion



feature selection

L1 $\rightarrow$ many $w = 0$

L2 $\rightarrow$ no $w=0$ many ≈ 0

Method 4, Data Augmentation

Discussion

- More training data can be created from the existing ones, for example, by translating or rotating the handwritten digits.

Hyperparameters

Discussion

- It is not clear how to choose the learning rate α, the stopping criterion ε, and the regularization parameters. $, \beta \ldots$
- For neural networks, it is also not clear how to choose the number of hidden layers and the number of hidden units in each layer.
- The parameters that are not parameters of the functions in the hypothesis space are called hyperparameters.

K Fold Cross Validation

Discussion

train on training set to find w, b
test on validation to compare performance
C, accuracy.

- Partition the training set into K groups.
- Pick one group as the validation set.
- Train the model on the remaining training set.
- Repeat the process for each of the K groups.
- Compare accuracy (or cost) for models with different hyperparameters and select the best one.

5 Fold Cross Validation Example

Discussion

- Partition the training set S into 5 subsets S_1, S_2, S_3, S_4, S_5

$$S_i \cap S_j = \emptyset \text{ and } \bigcup_{i=1}^{5} S_i = S$$

Iteration	Training	Validation
1	$S_2 \cup S_3 \cup S_4 \cup S_5$	S_1
2	$S_1 \cup S_3 \cup S_4 \cup S_5$	S_2
3	$S_1 \cup S_2 \cup S_4 \cup S_5$	S_3
4	$S_1 \cup S_2 \cup S_3 \cup S_5$	S_4
5	$S_1 \cup S_2 \cup S_3 \cup S_4$	S_5

get C on all training instances.

Leave One Out Cross Validation

Discussion

- If $\boxed{K = n}$, each time exactly one training instance is left out as the validation set. This special case is called Leave One Out Cross Validation (LOOCV).

Cross Validation, Part II

Quiz (Graded)

5

- March 2018 Midterm Q9 will repeat
- Consider the majority classifier that predict $\hat{y}$ = mode of the training data labels. What is the $\boxed{\text{2-fold}}$ cross validation accuracy (percentage of correct classification) on the following training set.

0 1 1 1 0

1

S_1 S_2

2/10 = 20%

x	1	2	3	4	5	6	7	8	9	10
y	1	1	0	1	1	0	0	1	0	0

- A: 0 percent, B: 10 percent, C: 20 percent
- D: 50 percent, E: 100 percent

correct.

$\hat{y}_{s_1} = 1$

Cross Validation, Part I

Quiz (Graded)

K = 10-fold CV

- March 2018 Midterm Q9
- Consider the majority classifier that predict $\hat{y}$ = mode of the training data labels. What is the LOOCV accuracy (percentage of correct classification) on the following training set.

put on midterm!

S_1 S_6

x	1	2	3	4	5	6	7	8	9	10
y	1	$\cancel{1}$	$\cancel{0}$	1	1	0	0	1	0	0

$\hat{y}_{S_1} = 0$

- A: 0 percent, B: 10 percent, C: 20 percent
- D: 50 percent, E: 100 percent

Multi-Class Classification

Discussion

- When there are K categories to classify, the labels can take K different values, $y_i \in \{1, 2, ..., K\}$.
- Logistic regression and neural network cannot be directly applied to these problems.

Method 1, One VS All

Discussion

0 vs not 0   1 vs not 1 ......

- Train a binary classification model with labels $y_i' = \mathbb{1}_{\{y_i = j\}}$ for each $j = 1, 2, ..., K$.
- Given a new test instance x_i, evaluate the activation function $a_i^{(j)}$ from model j.

$$\hat{y}_i = \arg\max_j a_i^{(j)}$$

- One problem is that the scale of $a_i^{(j)}$ may be different for different j.

Method 2, ~~One VS One~~

Discussion

0 vs 1 0 vs 2 0 vs 3 ...

- Train a binary classification model with for each of the $\frac{K(K-1)}{2}$ pairs of labels.
- Given a new test instance x_i, apply all $\frac{K(K-1)}{2}$ models and output the class that receives the largest number of votes.

$$\hat{y}_i = \arg\max_j \sum_{j' \neq j} \hat{y}_i^{(j \text{ vs } j')}$$

- One problem is that it is not clear what to do if multiple classes receive the same number of votes.

One Hot Encoding

no order

$y = 1, 2, 3, 4$.

- If y is not binary, use one-hot encoding for y.
- For example, if y has three categories, then

$$y_i \in \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

$y=1$ $y=2$ $y=3$

Method 3, Softmax Function

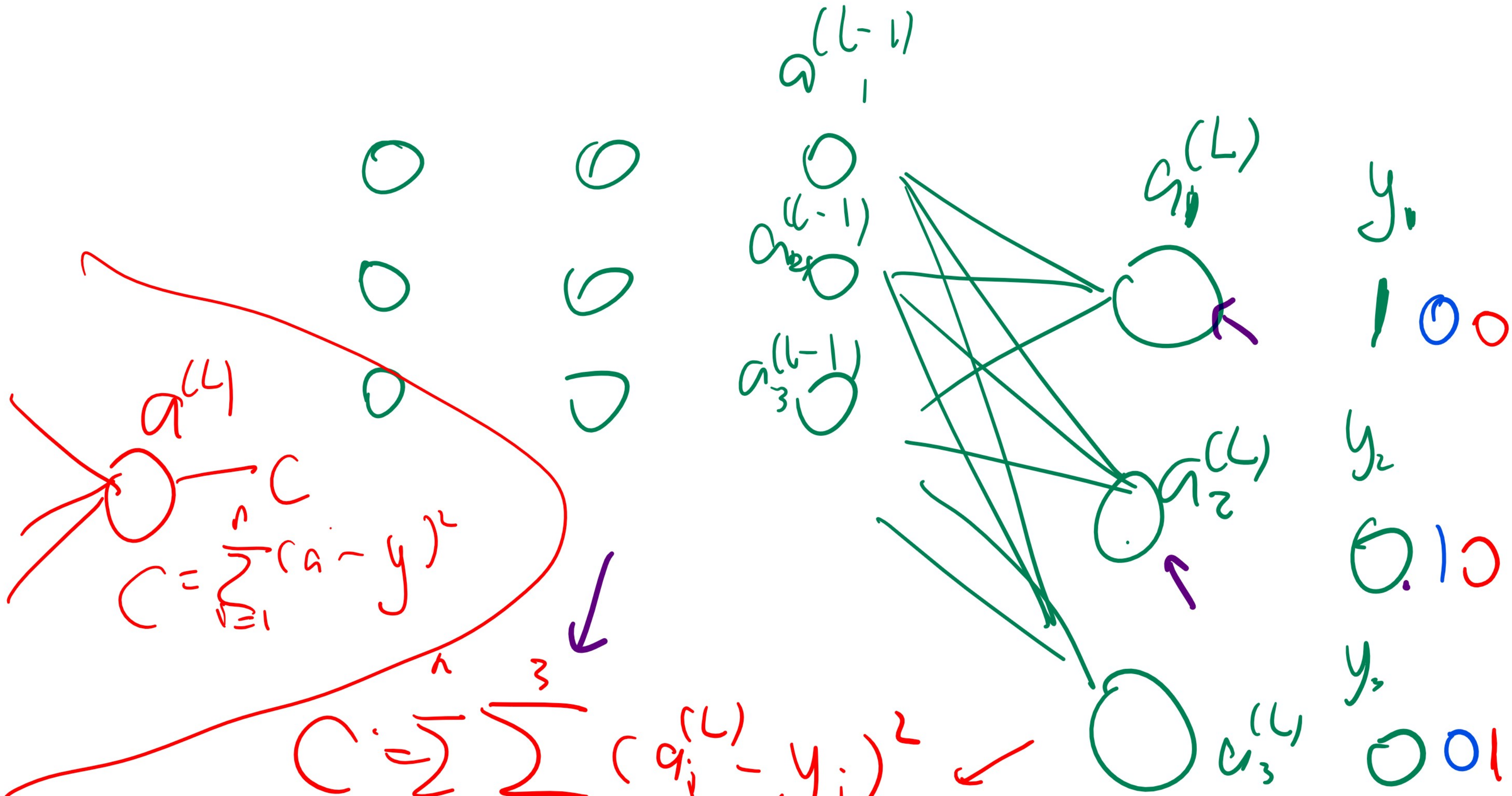
Discussion

- For both logistic regression and neural network, the last layer will have K units, a_{ij}, for $j = 1, 2, ..., K$ and the softmax function is used instead of the sigmoid function.

$$a_{ij} = g\left(w_j^T x_i + b_j\right) = \frac{\exp\left(w_j^T x_i + b_j\right)}{\underbrace{\sum_{j'=1}^{K} \exp\left(w_{j'}^T x_i + b_{j'}\right)}}, j = 1, 2, ..., K$$

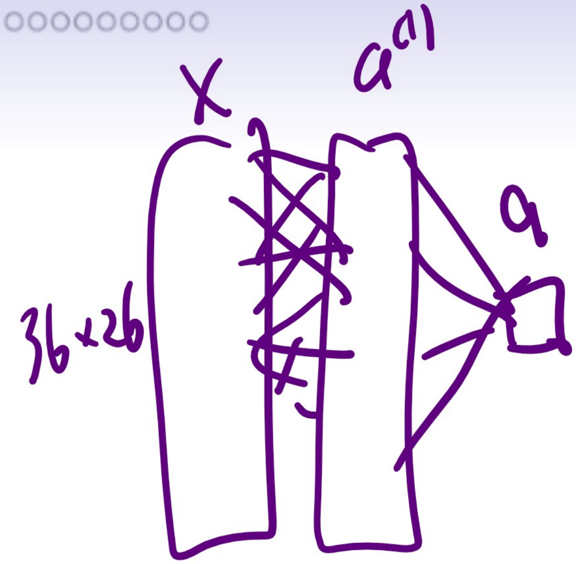
Softmax Function Diagram

Discussion



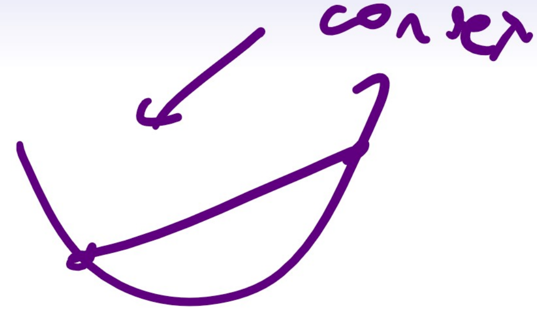
$$C = \sum_{i=1}^{n} \sum_{j=1}^{3} (a_j^{(L)} - y_j)^2$$

Same gradient descent.



Autoencoder

Discussion



- A multi-layer neural network with the same input and output $y_i = x_i$ is called an autoencoder.
- The hidden layers have fewer units than the dimension of the input m.
- The hidden units form an encoding of the input with reduced dimensionality.

Autoencode Diagram

Discussion

Generative Adversarial Network

Discussion

- Two competitive neural networks.

1. Generative network input random noise and output fake images.
2. Discriminative network input real and fake images and output label real or fake.

Generative Adversarial Network Diagram

Discussion