CSCE633 Exam 02

Lu Sun

October 2020

(1) (0.5 points) True/False: Using cross validation to select hyperparameters will guarantee that our model does not overfit.

Answer: False

When the train data set is systematically different from the test data set or the total train data set is too small, cross validation helps a little with overfitting.

(2) (0.5 points) For neural networks trained with stochastic gradient method, setting weights to 0 is an acceptable initialization.

Answer: False

No matter what gradient method is used, setting weights to 0 only results in all $\sigma = 0$. When using backward method to compute gradient, all the gradients keep to be 0 and nothing difference between new weights and old weights. Namely, for the following iterations, weights are always 0.

(3) (0.5 points) True/False: For logistic regression trained with stochastic gradient method, setting weights to 0 is an acceptable initialization.

Answer: True

The loss function(SSR) of logistic regression have convex function. It means that only one single optimal exists. No matter what the initial weights to be 0 or not, by gradient descent method, it can always converge to the optimal weights.

(4) (0.5 point) True/False: It is possible to represent a XOR function with a neural network without a hidden layer.

Answer: False

XOR function is multiple logical operations which is not the basic operation (AND, OR, NOT function). A hidden layer is necessary for XOR, the multiple logical operations.

(5) (0.5 points) True/False: A neural network with multiple hidden layers and sigmoid nodes can form non-linear decision boundaries.

Answer: True

Sigmoid function is nolinear function. With multiple hidden layer, the combination of sigmoid function and linear function can finally form non-linear decision function, resulting in non-linear decision boundaries.

(6) (0.5 points) Which of the following methods can achieve zero training error on any linearly separable dataset? (A) Support vector machines; (B) 3-Nearest Neighbor; (C) Linear perceptron; (D) Logistic regression.

Answer: ACD

A: Support vector machines are the combination of binary linear classification model which can achieve zero training error.

B: If a $'+1'$ point is closely around by 3 $'-1'$ point, misclassification will occur.

C: Linear perceptron in binary classification problem, will achieve zero training error on linearly separable data set in finite steps.

D: Logistic regression is a linear classification model in essential, which can achiever zero training error on linearly separable data set.

(7) (0.5 points) We use a support vector machine (SVM) to classify the following samples. The samples from class +1 are represented in orange color, while the samples from class -1 are plot in blue. The decision boundary is represented by the green line. How many support vectors are there?

Answer: 5

According to the green line, decision boundary, the support vectors should be the 4 closest blue points on the left of the line and 1 closet orange point on the right of the line. $4 + 1 = 5$.

(8) (0.5 points) We use a support vector machine (SVM) to classify the following samples. The samples from class +1 are represented in orange color, while the samples from class -1 are plot in blue. The decision boundary is represented by the green line. What condition holds for the soft error S? $(A)S > 0; (B)S = 0; (C)S < 0$

Answer: (A)

According to the green line, correct classified points are the right 3 orange points and left 4 blue points, whose $\zeta = 0$. No points are inside margin. One left orange point is misclassification, whose $\zeta > 0$. By the definition of $S = \sum_n \zeta_n$, we have $S > 0$.

(9) (1 point) True/False: The following multi-layer perceptron (i.e., feedforward neural network) with linear activation function at its hidden nodes and sigmoid activation at its output is equivalent to a logistic regression model with the same number of features.

Answer: True

Suppose the linear activation function $f(x) := ax + b$, here a, b are constant. Then the perceptron in picture can be represented by the following formula:

$$h(X_1, X_2) = \sigma[w_5^T f(w_1^T X_1 + w_3^T X_2) + w_6^T f(w_2^T X_1 + w_4^T X_2)]$$

$$h(x_1, X_2) = \sigma[(w_5^T a w_1^T + w_6^T a w_2^T)X_1 + (w_5^T a w_3^T + w_6^T a w_4^T)X_2 + (w_6 + w_5)^T b]$$

$$h(x_1, X_2) = \sigma[(W_{new,1}X_1 + W_{new,2}X_2 + W_{new,0}]$$

Finally, corresponding to logistic regression model.

(10) (0.5 point) When performing hyper-parameter tuning of a neural network, why would random hyper-parameter search be more likely to converge compared to grid-search?

Answer: In the setting of neural network, more hyper-parameter and data set are given than other model. But not all the hyper-parameter are equally important. Grid-search also can't find the optimal hyper-parameter in this setting. However, in random hyper-parameter search, as random values are selected at each instance, it is highly likely that the whole of action space has been reached because of the randomness and it can end up being trained on the optimised parameters without any aliasing. While, it takes a huge amount of time to cover every aspect of the combination during grid search. In general, the chances of finding the optimal parameter are comparatively higher in random search because of the randomise than grid-search.

(11) (0.5 point) In the following, we provide the original image I (top) and the resulting images I1 and I2 (bottom) that have been transformed through a convolutional filter. What type of filter was applied to each of the resulting images I1 and I2?

Answer: l1: vertical filter, l2 horizontal filter

l1 has boundary on the left and right of the figure, and shows the vertical outline of the original figure.

l2 has boundary on the top and bottom of the figure, and shows the horizontal outline of the original figure.

(12) (0.5 point) A 10x10 image is the input to a convolutional neural network (CNN). The first hidden layer of the CNN is comprised of 4 3x3 filters. Assuming zero-padding and bias, what is the dimensionality of the output from the first hidden layer and the number of parameters that are estimated from this layer?

Answer: Output dimension: $10 \times 10 \times 4$, Number of parameters: 40

Because of zero-padding, the output shape won't change. Hidden layer has 4 filers, so the output dimension should be $10 \times 10 \times 4$.

Because of the existence for bias, number of parameters equals $(3 \times 3+1) \times 4 = 40$

(13) (0.5 point) A 10x10 image is the input to a convolutional neural network (CNN). The first hidden layer of the CNN is comprised of 4 3x3 filters. Assuming bias but not zero-padding, what is the dimensionality of the output from the first hidden layer and the number of parameters that are estimated from this layer?

Answer: Output dimension: $8 \times 8 \times 4$, Number of parameters: 40

Because of none zero-padding, the output shape changes to be $10-(3-1) = 8$. Hidden layer has 4 filers, so the output dimension should be $8 \times 8 \times 4$.

Because of the existence for bias, number of parameters equals $(3 \times 3+1) \times 4 = 40$

(14) (1 point) We perform one update step with an iterative optimization algorithm with starting sample depicted by the blue asterisk and end sample depicted by the red asterisk. The arrows starting from the starting sample depict the direction of the update step. Please explain which of the two arrows is more likely to correspond to stochastic gradient descent and which is more likely to correspond to RMSprop.

Answer: Black arrow: stochastic gradient descent, Red arrow: RMSprop

In Stochastic gradient descent method, the gradient $g_{sgd} = \nabla_\theta J(\theta; x_i, y_i)$. Namely, an update for all parameters θ at once as every parameter θ_i used the same learning rate. So the direction should perpendicular to the initial line. Only black arrow perpendicular to the line.

In RMSprop method, the gradient $g_{RMSprop} = (\nabla_\theta J(\theta_i))_i$. Namely, an update for all parameters θ at once as every parameter θ_i used the different learning rate. So the direction shouldn't perpendicular to the initial line. Only red arrow satisfies the property.

(15) (2 points) Maximum likelihood estimate

Answer: $\theta^* = \frac{1}{2}$

Assume $f(\theta) := P(\theta|\{1,1,1,2,2,3,3,4,4,4\})$, then $f(\theta)$ has the following equation:

$$\begin{aligned} f(\theta) &= p(\theta = 1)^3 p(\theta = 2)^2 p(\theta = 3)^2 p(\theta = 4)^3 \\ &= (\tfrac{\theta}{2})^3 (\tfrac{\theta}{2})^2 (\tfrac{1-\theta}{3})^2 (\tfrac{2-2\theta}{3})^3 \\ &= 2^{-2} 3^{-5} (\theta)^5 (1-\theta)^5 \end{aligned}$$

Assume $L(\theta) := \log(f(\theta)) = 5\log(\theta) + 5\log(\theta) - 2\log 2 - 5\log 3$. The corresponding maximum likelihood estimation is shown as follow:

$$\max_\theta f(\theta) \Longleftrightarrow \max_\theta L(\theta)$$

Then,

$$\frac{\partial L(\theta)}{\partial \theta} = \frac{5}{\theta} + \frac{-5}{1-\theta} = \frac{5(1-2\theta)}{\theta(1-\theta)}, \theta \in (0,1)$$

$$\frac{\partial^2 L(\theta)}{\partial \theta^2} = \frac{-10\theta(1-\theta) - 10\theta(1-2\theta)}{\theta^2(1-\theta)^2} = \frac{-10(2-3\theta)}{\theta(1-\theta)^2}, \theta \in (0,1)$$

By setting $\frac{\partial L(\theta)}{\partial \theta} = 0$, we have $\theta^* = \frac{1}{2}$ and $\frac{\partial^2 L(\theta)}{\partial \theta^2}|_{\theta=\frac{1}{2}} = -40 < 0$. Namely, $\theta^* = \frac{1}{2}$ is the maximum likelihood estimate for the initial problem.