Unit 6 Softmax Regression Part 02 Softmax Classifier Training

TFIP-AI Artificial Neural Networks and Deep Learning

Softmax Regression Cost Function

We now describe the cost function that we'll use for softmax regression. In the equation below, $1\{\cdot\}$ is the "'indicator function,"' so that $1\{\text{a true statement}\} = 1$, and $1\{\text{a false statement}\} = 0$. For example, $1\{2 + 2 = 4\}$ evaluates to 1; whereas $1\{1 + 1 = 5\}$ evaluates to 0. Our cost function will be:

$$J(\theta) = -\left[\sum_{i=1}^{m}\sum_{k=1}^{K} 1\left\{y^{(i)} = k\right\} \log \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})}\right]$$

Softmax Regression Cost Function cont...

Notice that this generalizes the logistic regression cost function, which could also have been written:

$$
\begin{aligned}
J(\theta) &= -\left[\sum_{i=1}^{m} (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) + y^{(i)} \log h_\theta(x^{(i)}) \right] \\
&= -\left[\sum_{i=1}^{m} \sum_{k=0}^{1} 1 \left\{ y^{(i)} = k \right\} \log P(y^{(i)} = k | x^{(i)}; \theta) \right]
\end{aligned}
$$

Softmax Regression Cost Function cont...

The softmax cost function is similar, except that we now sum over the K different possible values of the class label. Note also that in softmax regression, we have that

$$P(y^{(i)} = k|x^{(i)}; \theta) = \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})}$$

Softmax Regression Cost Function cont...

We cannot solve for the minimum of $J(\theta)$ analytically, and thus as usual we'll resort to an iterative optimization algorithm. Taking derivatives, one can show that the gradient is:

$$\nabla_{\theta^{(k)}} J(\theta) = -\sum_{i=1}^{m} \left[x^{(i)} \left(1\{y^{(i)} = k\} - P(y^{(i)} = k | x^{(i)}; \theta) \right) \right]$$

Recall the meaning of the "$\nabla_{\theta^{(k)}}$" notation. In particular, $\nabla_{\theta^{(k)}} J(\theta)$ is itself a vector, so that its j-th element is $\frac{\partial J(\theta)}{\partial \theta_{lk}}$ the partial derivative of $J(\theta)$ with respect to the j-th element of $\theta^{(k)}$.

Armed with this formula for the derivative, one can then plug it into a standard optimization package and have it minimize $J(\theta)$.

Relationship to Logistic Regression

In the special case where $K = 2$, one can show that softmax regression reduces to logistic regression. This shows that softmax regression is a generalization of logistic regression. Concretely, when $K = 2$, the softmax regression hypothesis outputs

$$h_\theta(x) = \frac{1}{\exp(\theta^{(1)\top}x) + \exp(\theta^{(2)\top}x^{(i)})} \begin{bmatrix} \exp(\theta^{(1)\top}x) \\ \exp(\theta^{(2)\top}x) \end{bmatrix}$$

Relationship to Logistic Regression cont...

Taking advantage of the fact that this hypothesis is overparameterized and setting $\psi = \theta^{(2)}$, we can subtract $\theta^{(2)}$ from each of the two parameters, giving us

$$
\begin{aligned}
h(x) &= \frac{1}{\exp((\theta^{(1)} - \theta^{(2)})^\top x^{(i)}) + \exp(\vec{0}^\top x)} \left[\exp((\theta^{(1)} - \theta^{(2)})^\top x) \; \exp(\vec{0}^\top x) \right] \\
&= \begin{bmatrix} \frac{1}{1+\exp((\theta^{(1)}-\theta^{(2)})^\top x^{(i)})} \\ \frac{\exp((\theta^{(1)}-\theta^{(2)})^\top x)}{1+\exp((\theta^{(1)}-\theta^{(2)})^\top x^{(i)})} \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{1+\exp((\theta^{(1)}-\theta^{(2)})^\top x^{(i)})} \\ 1 - \frac{1}{1+\exp((\theta^{(1)}-\theta^{(2)})^\top x^{(i)})} \end{bmatrix}
\end{aligned}
$$

Relationship to Logistic Regression cont...

Thus, replacing $\theta^{(2)} - \theta^{(1)}$ with a single parameter vector θ', we find that softmax regression predicts the probability of one of the classes as $\frac{1}{1+\exp(-(\theta')^\top x^{(i)})}$, and that of the other class as $1 - \frac{1}{1+\exp(-(\theta')^\top x^{(i)})}$, same as logistic regression.

Multi-class classification -- Training a softmax classifier

Understanding softmax

$$g^{[L]}(Z^{[L]}) = \begin{bmatrix} \frac{e^5}{e^5+e^2+e^{-1}+e^3} \\ \frac{e^2}{e^5+e^2+e^{-1}+e^3} \\ \frac{e^{-1}}{e^5+e^2+e^{-1}+e^3} \\ \frac{e^3}{e^5+e^2+e^{-1}+e^3} \end{bmatrix} = \begin{bmatrix} 0.842 \\ 0.042 \\ 0.002 \\ 0.114 \end{bmatrix}$$

"Softmax" vs "Hard Max [1, 0, 0, 0]"

Softmax regression generalizes logistic regression to C classes rather than just two classes.

Multi-class classification -- Training a softmax classifier cont...

Softmax regression generalizes logistic regression to C classes rather than just two classes.

If C=2, softmax reduces to logistic regression.

Loss function

$$y^{(1)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

This represents a "cat".

$$y_2^{(1)} = 1, y_1^{(1)} = y_3^{(1)} = y_4^{(1)} = 0$$

Loss function cont...

$$a^{[L](1)} = \hat{y}^{(1)} = \begin{bmatrix} 0.3 \\ 0.2 \\ 0.1 \\ 0.4 \end{bmatrix}$$

$$L(\hat{y}, y) = -\sum_{j=1}^{4} y_j \log \hat{y}_j = -y_2 \log \hat{y}_2 = -\log \hat{y}_2$$

Loss function cont...

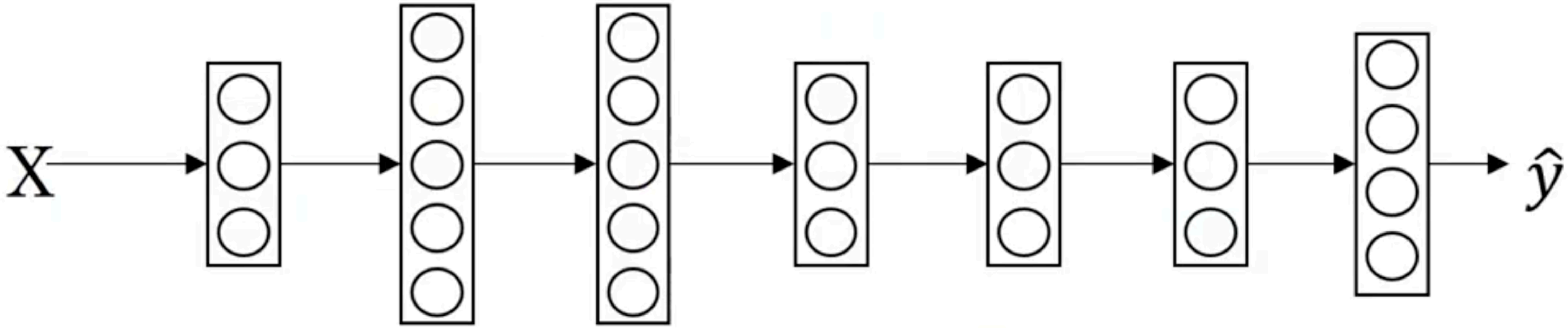
$$Y = \left[y^{(1)} y^{(2)} ... y^{(m)} \right]$$

$$= \begin{bmatrix} 0 & 0 & 1 & \\ 1 & 0 & 0 & \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & \end{bmatrix}$$

$$\hat{Y} = \left[\hat{y}^{(1)} \hat{y}^{(2)} ... \hat{y}^{(m)} \right]$$

$$= \begin{bmatrix} 0.3 & \\ 0.2 & \\ 0.1 & \cdots\cdots \\ 0.4 & \end{bmatrix}$$

Gradient descent with softmax



Forward propagation step.

$$Z^{[L]} \longrightarrow a^{[L]} = \hat{y} \longrightarrow L(\hat{y}, y)$$

Backward propagation step.

$$dZ^{[L]} = \hat{y} - y \longleftarrow \frac{\partial J}{\partial Z^{[L]}}$$