October 19, 2017 Lecture 11, Oct. 11, 2017

Back Propagation

Lecturer: Justin Domke *Scribe: Boya Ren*

1 Summary

Last time we stopped at forward propagation and back propagation. Today we will continue discussing back propagation and its application in machine learning.

2 Forward Propagation and Back Propagation

2.1 Forward Propagation

```
1. For $i = n+1, n+2, \cdots, N$
       $x_i \leftarrow g_i(x_{Pa(i)})$
2. Return $x_N$
```

2.2 Back Propagation V1

```
1. $\frac{df}{dx_N} = 1$
2. For $i = N-1, N-2, \cdots, 1$
       $\frac{df}{dx_i} \leftarrow \sum_{j:i \in Pa(i)} \frac{df}{dx_j}\frac{dg_j}{dx_i}$
```

2.3 Back Propagation V2

```
1. $\frac{df}{dx_N} \leftarrow 1$
2. $\frac{df}{dx_1} = \frac{df}{dx_2} = \cdots = \frac{df}{dx_{N-1}} \leftarrow 0$
3. For $j = N, N-1, \cdots, n+1$
       For all $i \in Pa(j)$
           $\frac{df}{dx_i} \leftarrow \frac{df}{dx_i} + \frac{df}{dx_j}\frac{dg_i}{dx_i}$
```

2.4 Discussions

These two versions of back propagation are actually the same, one from the children point of view, and the other from the parents point of view. So why is the algorithm so great? There are mainly two reasons.

- Just need to figure out derivatives for each type of function once. Then combine arbitrarily.
- Compute ∇f takes around the same time as f itself.

3 Using Auto-Differentiation for ML

3.1 Linear - Least - Squares Regression

For single linear output linear regression, we can write its prediction function as $f(x) = B^T x$ and loss function $L(y, f) = (y - f)^2$. Then we can picture $L(y, f)$ as a single expression graph as below.

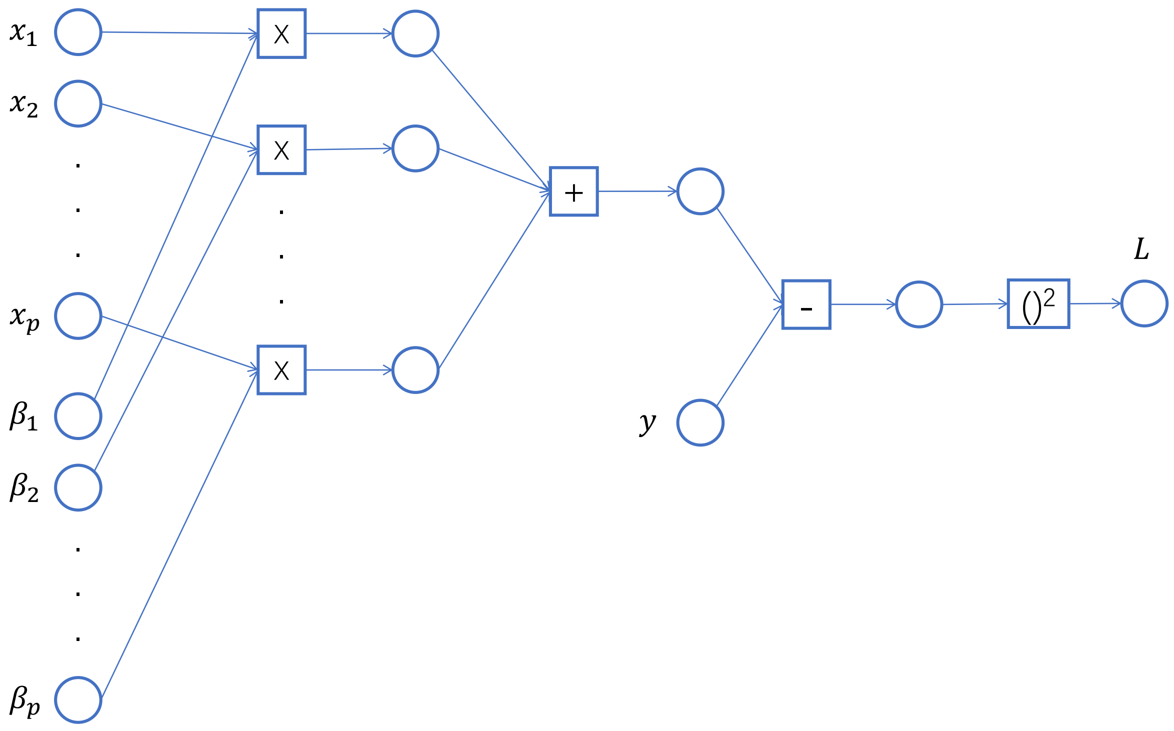

Figure 3.1: Expression Graph for Linear-Least-Squares Regression

Then what about a multiple output regression? In this case, the function $f(x)$ is $R^N \rightarrow R^K$ supposing K outputs. The graph representation can be shown in **Fig. 3.2**.

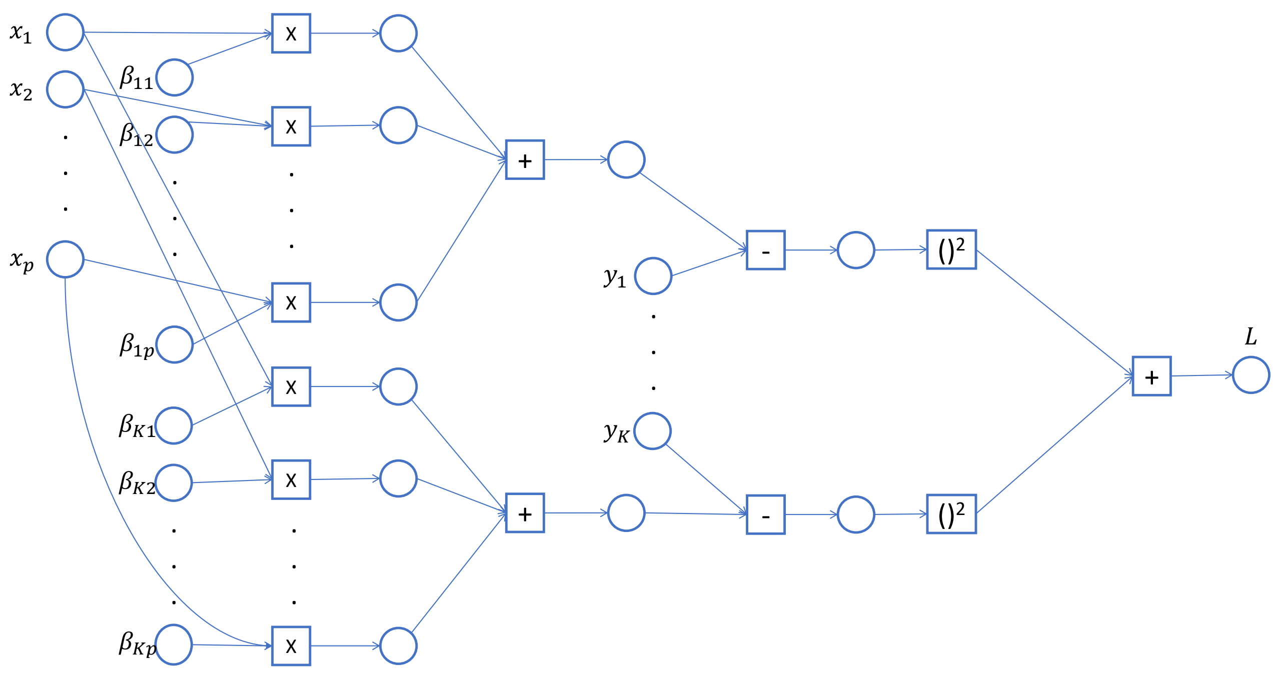

Figure 3.2: Expression Graph for Multiple Output Regression

3.2 Compressed Visual Notations

The above graph presentation looks tedious and hard to draw. A better way, which initially might be a little bit confusing, is to suppress parameters, drop function nodes, so that the graph looks simpler. For example, the multiple - output linear regression can be expressed in **Fig. 3.3**.

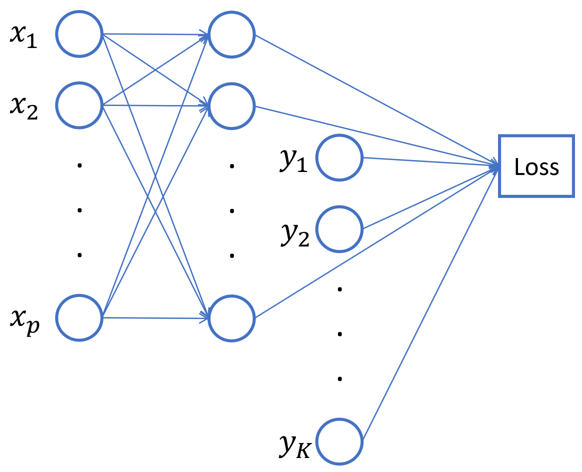

Figure 3.3: Compressed Expression for Linear Regression

3.3 Neural Network

Then how can we improve the classification model from linear regression? Inspired by the graph notation, the first idea is, add more layers to the network, as illustrated in **Fig. 3.4**.

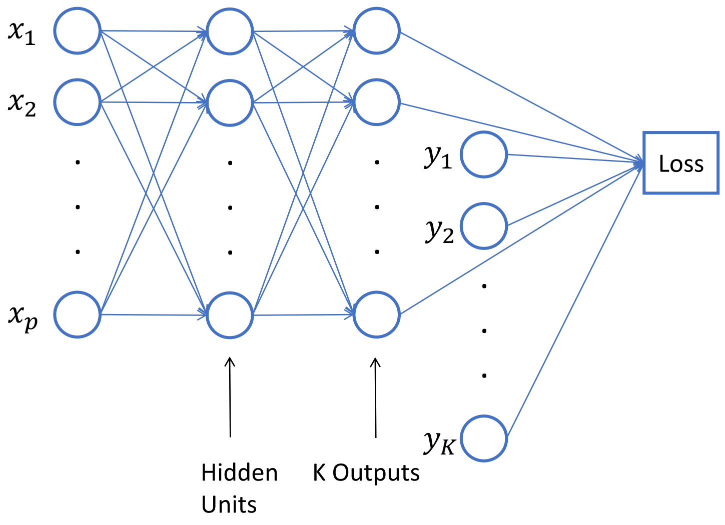

Figure 3.4: A Network with One Hidden Layer

Is this a good idea? Not really, because several linear transforms together is still a linear transform. Suppose the above network can be represented as $f(x) = VWX$. Then it is exactly the same as $f(x) = B^TX$ if $B^T = VW$.

A typical solution to this problem is to add some nonlinearity between layers. To be more precise, the network expression can be written as $f(x) = V\sigma(WX)$, where $\sigma(\cdot)$ is a componentwise nonlinear function. Some common choices are,

- "sigmoid": $\sigma(s) = \frac{1}{1+\exp(-s)}$
- "tanh": $\sigma s = \tanh(s)$
- "ReLU": $\sigma(s) = \max(0, s)$

3.4 Network for HW2

Your network in HW2 can be expressed as $f(x) = c + V\sigma(b + WX)$, where $x \in R^p$, $W \in R^{M\times p}$, $b \in R^M$, $\sigma = \tanh$, $V \in R^{K\times M}$ and $c \in R^K$. The loss function can be written as $L(y, f) = -f_y + \log\sum_{k=1}^{K}\exp(f_k)$. It can be interpreted as, if only f_y is one and the rest are zero, the loss would be zero, and otherwise, loss will be high. In order to implement back propagation, you first have to implement the derivatives, dL/dw, dL/db, dL/dv, dL/dc. We will first write down the analytical results and explain in the next lecture.

$$\frac{dL}{df} = -\hat{e}_y + g(f),\ \ g(T)_i = \frac{\exp(T_i)}{\sum_{k=1}^{K}\exp(T_k)}$$

$$\frac{dL}{dc} = \frac{dL}{df},\ \frac{dL}{dv} = \frac{dL}{df} h^T$$

$$\frac{dL}{db} = \sigma'(b + WX) \odot (V^T \frac{dL}{df})$$

$$\frac{dL}{dW} = \left(\sigma'(b + WX) \odot (V^T \frac{dL}{df}) \right) X^T$$

Here $\odot$ is element-wise multiplication.