# Likelihood Machine Learning Seminar II

## Lecture 7: Neural Network

Likelihood Lab
XingYu Fu

## 1. Forward Propagation (Inference)

Neural Network is a non-linear machine learning algorithm that is famous for its strong predictive power and its black-box property. It contains multiple layers, each of which contains a weight matrix and a bias vector and each row of the matrixes is termed as *neuron*. In each layer, the input vector $I \in R^{n \times 1}$ of the layer is manipulated in two phases:

**- Linear Transformation**

$$O := WI + b$$

, where $W \in R^{m \times n}$ is the weight matrix, $b \in R^{m \times 1}$ is the bias vector and $O \in R^{m \times 1}$ is the linear output of this layer.

**- Non-Linear Transformation**

$$\hat{O} := f(O)$$

, where $f$ is a non-linear function that applied to $O$ term-wise and $\hat{O}$ is the final non-linear output of this layer. The non-linear function is also called as *activation* function in neural network context.

The non-linear output of the current layer is passed to the next layer as input and this process continues until there is no more next layer. Formally, we can write as:

$$I_{k+1} = \hat{O}_k$$

, where $k$ is the index of current layer.

## 2. Backward Propagation (Training)

Similar to logistic regression, we train neural network by minimizing a loss function $L$, which is differentiable with respect to the model parameters, through gradient descent algorithm. Due to the hierarchical structure of the model, we derive the gradient or partial derivative of the loss function by *chain rule*.

For the $k^{th}$ layer, we can see that:

$$\frac{\partial L}{\partial W_k} = \frac{\partial L}{\partial O_N}\left(\frac{\partial O_N}{\partial I_N}\frac{\partial I_N}{\partial O_{N-1}}\right)\ldots\left(\frac{\partial O_{k+1}}{\partial I_{k+1}}\frac{\partial I_{k+1}}{\partial O_k}\right)\frac{\partial O_k}{\partial W_k}$$

, where $N$ is the total number of layers and, by section 1, we can see that:

$$\frac{\partial O_j}{\partial W_j} = I_j^T$$

$$\frac{\partial O_j}{\partial I_j} = W_j$$

$$\frac{\partial I_j}{\partial O_{j-1}} = f_{j-1}^{'}(O_{j-1})$$

The value of $\frac{\partial L}{\partial O_N}$ is determined by the format of loss function. For example, say $L = \frac{1}{2}||\hat{O}_N - y||^2$, then by chain rule, we can see that:

$$\frac{\partial L}{\partial O_N} = (\hat{O}_N - y)f_N^{'}(O_N)$$

Now, we have derived all formulas needed to compute the derivative of loss function with respect to model parameters.

In practice, you need to compute the formulas backwardly so that we can reuse some computation results to accelerate the training.