
# INFO 7375 - Neural Networks & AI

Homework to Chapter - 8

Submitted By:

Abdul Haseeb Khan
NUID: 002844724
khan.abdulh@northeastern.edu

## What is the reason for softmax?

Softmax is used to handle multi-class classification by normalizing a model's raw outputs into a probability distribution over K classes, which makes the outputs easier to interpret as class membership probabilities. Softmax is an activation function that scales numbers or logits into probabilities and is usually placed as the last layer to output a vector whose elements sum to 1.

It is often used as the last activation function of a neural network to normalize the output to a probability distribution over predicted output classes, so each component can be read as $P(Y = k \mid X)$ for the corresponding class k. The softmax activation function makes the neural network's outputs easier to interpret by transforming raw outputs into a vector of probabilities that sum to one over the input classes. In multi-class settings, the hypothesis outputs a K-dimensional vector whose elements sum to 1, providing estimated probabilities for each of the K classes.

## What is softmax and how does it works?

Softmax regression, or multinomial logistic regression, is a generalization of logistic regression to handle multiple classes where $Y^{(i)} \in \{1,\ldots,K\}$ rather than binary labels. As an activation function, softmax scales the model's logits into probabilities so that the output vector v contains a probability for each possible outcome and the probabilities in v sum to one. It is related to the argmax function, acting as a softer version that yields probability-like outputs rather than a hard winner-take-all 0 or 1 selection.

Given an input
X
X, the softmax hypothesis estimates P(Y=k|X) for each class $k = 1,\ldots,K$, producing a K-dimensional probability vector that sums to 1 across classes. Concretely, the hypothesis has the form $h_\theta(X)_k = \frac{exp(\theta^{(k)\top} X)}{\sum_{j=1}^{K} exp(\theta^{(j)\top} X)}$, where the denominator normalizes the distribution so the probabilities sum to one. In implementation, it is convenient to represent θ as an n×K matrix formed by concatenating $\theta^{(1)}, \ldots, \theta^{(K)}$ into columns, reflecting the parameters that map inputs to the K outputs. For multi-class image recognition such as dog, cat, horse, or cheetah, a final fully connected layer outputs logits that a softmax layer transforms into class probabilities P, which serve as the model's predictions for each class. During training with softmax, target labels are vectors with 1 for the target class and 0 for all other classes, and the associated cost function sums over the K possible class values using the indicator function 1{·} where 1{true}=1 and 1{false}=0. One noted limitation is that softmax can cause output values to become extremely large or small, which may make gradient-based optimization difficult and lead to slow convergence or high variance in training.