The first hidden unit (the Dense layer in Keras) is a function H=f1(**X,W1,b1**) =activation function(**XW1** + **b1**)

X: the input data.

W1: weight matrix of the first unit

b1: bias vector

H: resulting vector

It firstly applies the linear transformation **XW1** + **b1** then applies the activation function **relu** (**rectified linear unit**) on the resulting vector H.

relu(x)=max{0,x}

Why **relu**?

- It can output zero which facilitates representation learning by reducing the dimension of input data
- It still retains an approximate linear behavior
- It makes the function f(**X,W,b**) a non-linear function

The output unit transforms Z=W2H+B2 then makes use of the **softmax** activation function:

z is a vector in Z and z_i is component of z.

P~: estimated probability distribution

softmax(z_i)=$\frac{e^{z_i}}{\sum_1^n e^{z_j}}$ where $z_i = \log P\sim (y = i|x)$

It outputs a 10D vector indicating the probabilities with respect to 10 labels given a data point.

The loss function **categorical-crossentropy** is based on maximum likelihood. Unregularized maximum likelihood pushes the model to learn parameters so that softmax predicts the empirical distribution of the train data.

The optimizer **rmsprop** is an upgraded version of stochastic gradient descent.