Leren — Homework 5

Chapter 11, Alpaydin

Tim Stolp (11848782)

Deadline: 23:59 December 16th, 2018

Artificial neural networks are machine learning models. These models can be trained with *gradient descent*: small steps in the direction of the gradient are taken, to minimize/maximize a loss/likelihood function. Therefore, to train a model, we need two things: how to compute the prediction (forward), and how to compute the gradient (backward).

1 Multinomial Regression / Multiclass Discrimination

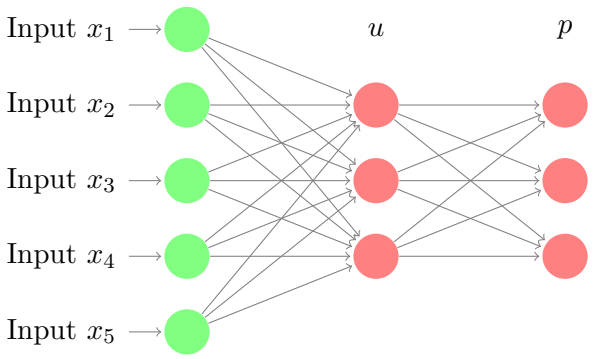

Figure 1: Multinomial Regression

Multinomial regression models make predictions for multi-class problems. The likelihood for such a classification problem, given a dataset $\{(\mathbf{x}^{(i)}, t^{(i)})\}_{i=1}^{N}$, is written as:

$$\prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)})$$

And the log-likelihood as:

$$\mathcal{L} = \sum_{i=1}^{N} \log p(t^{(i)}|\mathbf{x}^{(i)})$$

For now, we will focus on the likelihood of a single datapoint and omit the $^{(i)}$ notation. Also we let $p(t|\mathbf{x}) = p_t$:

$$\ell = \log p(t|\mathbf{x}) = \log p_t$$

For every class $j = 1, \ldots, C$, a value u_j is put through a softmax to obtain a probability distribution over classes. The probability that $\mathbf{x}$ belongs to class j, $p_j = p(j|\mathbf{x})$, is defined as:

$$p_j = \frac{\exp(u_j)}{\sum_{k=1}^{C} \exp(u_k)}$$

this ensures that the sum of probabilities $\sum_j p_j$ is equal to 1.

And we can write $\log p_j$ as:

$$\log p_j = u_j - \log \sum_{k=1}^{C} \exp u_k$$

In this first part we choose a simple linear model for u_j. In this equation $\mathbf{w}_j$ and b_j are parameters.:

$$u_j(\mathbf{x}) = \mathbf{w}_j^T \mathbf{x} + b_j$$

These equations together define the prediction/forward computation of the multinomial regression model. To optimize the likelihood, we will use a gradient method. As such, gradients from the likelihood to the parameters are needed: $\frac{\partial \ell}{\partial \mathbf{w}_j}$ and $\frac{\partial \ell}{\partial b_j}$, for all j. To obtain these, we use the chain rule and find expressions for $\frac{\partial \ell}{\partial u_j}$, $\frac{\partial u_j}{\partial \mathbf{w}_j}$ and $\frac{\partial u_j}{\partial b_j}$.

(a) Show that $\frac{\partial \ell}{\partial u_j}$ is equal to $\left\{ \begin{array}{ll} 1 - \frac{\exp u_j}{\sum_{k=1}^{C} \exp u_k}, & \text{if } j = t \\ - \frac{\exp u_j}{\sum_{k=1}^{C} \exp u_k}, & \text{if } j \neq t \end{array} \right\}$

If $j = t$:
$\frac{\partial \ell}{\partial u_j} = \frac{\partial}{\partial u_j}(u_t - \log \sum_{k=1}^{C} \exp(u_k))$

$= 1 - \frac{1}{\sum_{k=1}^{C} \exp(u_k)} \cdot \exp(u_j)$

$= 1 - \frac{\exp(u_j)}{\sum_{k=1}^{C} \exp(u_k)}$

If $j \neq t$:
$\frac{\partial \ell}{\partial u_j} = \frac{\partial}{\partial u_j}(u_t - \log \sum_{k=1}^{C} \exp(u_k))$

$= -\frac{1}{\sum_{k=1}^{C} \exp(u_k)} \cdot \exp(u_j)$

$= -\frac{\exp(u_j)}{\sum_{k=1}^{C} \exp(u_k)}$

(b) Give the derivative for $\frac{\partial u_j}{\partial \mathbf{w}_j}$

According to the rule: $\frac{\partial x^T y}{\partial x} = \frac{\partial y^T x}{\partial x} = y$,

We get:
$\frac{\partial u_j}{\partial \mathbf{w}_j} = \frac{\partial}{\partial \mathbf{w}_j}(\mathbf{w}_j^T \mathbf{x} + b_j) = x + 0 = x$

(c) Give the derivative for $\frac{\partial u_j}{\partial b_j}$

$\frac{\partial u_j}{\partial b_j} = \frac{\partial}{\partial b_j}(\mathbf{w}_j^T\mathbf{x} + b_j) = 0 + 1 = 1$

(d) Give an expression for the gradient $\frac{\partial \ell}{\partial \mathbf{w}_j}$ and $\frac{\partial \ell}{\partial b_j}$

If $j = t$:
$\frac{\partial \ell}{\partial \mathbf{w}_j} = \frac{\partial}{\partial \mathbf{w}_j}(u_t - \log\sum_{k=1}^{C}\exp(u_k))$
$= \frac{\partial}{\partial \mathbf{w}_j}(\mathbf{w}_t^T\mathbf{x} + b_t - \log\sum_{k=1}^{C}\exp(u_k))$

$= x - \frac{\partial}{\partial \mathbf{w}_j}(\log\sum_{k=1}^{C}\exp(\mathbf{w}_k^T\mathbf{x} + b_k))$

$= x - \frac{\exp(\mathbf{w}_j^T\mathbf{x}+b_j)\cdot x}{\sum_{k=1}^{C}\exp(\mathbf{w}_k^T\mathbf{x}+b_k)}$

If $j \neq t$:
$\frac{\partial \ell}{\partial \mathbf{w}_j} = \frac{\partial}{\partial \mathbf{w}_j}(u_t - \log\sum_{k=1}^{C}\exp(u_k))$

$= -\frac{\exp(\mathbf{w}_j^T\mathbf{x}+b_j)\cdot x}{\sum_{k=1}^{C}\exp(\mathbf{w}_k^T\mathbf{x}+b_k)}$

If $j = t$:
$\frac{\partial \ell}{\partial b_j} = \frac{\partial}{\partial b_j}(u_t - \log\sum_{k=1}^{C}\exp(u_k))$
$= \frac{\partial}{\partial b_j}(\mathbf{w}_t^T\mathbf{x} + b_t - \log\sum_{k=1}^{C}\exp(u_k))$

$= \frac{\partial}{\partial b_j}(\mathbf{w}_t^T\mathbf{x} + b_t) - \frac{\partial}{\partial b_j}(\log\sum_{k=1}^{C}\exp(\mathbf{w}_k^T\mathbf{x} + b_k))$

$= 1 - \frac{\exp(\mathbf{w}_j^T\mathbf{x}+b_j)}{\sum_{k=1}^{C}\exp(\mathbf{w}_k^T\mathbf{x}+b_k)}$

If $j \neq t$:
$\frac{\partial \ell}{\partial b_j} = \frac{\partial}{\partial b_j}(u_t - \log\sum_{k=1}^{C}\exp(u_k))$

$= -\frac{\exp(\mathbf{w}_j^T\mathbf{x}+b_j)}{\sum_{k=1}^{C}\exp(\mathbf{w}_k^T\mathbf{x}+b_k)}$

(e) Explain why the derivative $\frac{\partial u_j}{\partial \mathbf{w}_i}$ is 0, if $i \neq j$.

There is no w_i and $i \neq j$ which makes everything a constant. The derivative of a constant is 0.

2 Neural Networks

In the previous exercise, the unnormalized probability u_j was a linear model of $\mathbf{x}$. In neural networks, multiple linear models can be stacked and alternated with activations functions:

We let u_j be a linear function of hidden layer $\mathbf{h}$.

$$u_j = \mathbf{w}_j^T\mathbf{h} + b_j$$

The hidden layer $\mathbf{h}$ is obtained by applying a sigmoid activation function on the pre-activations

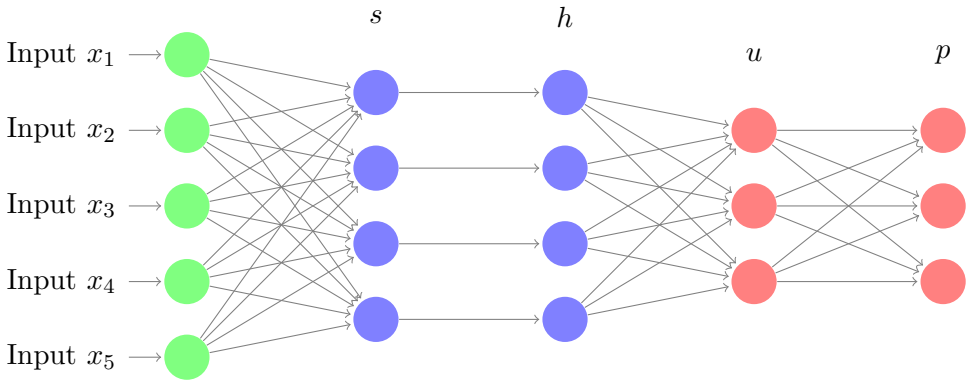

Figure 2: Neural Network

s.

$$h_k = \sigma(s_k) = \frac{1}{1 + \exp(-s_k)}$$

And the pre-activations $\mathbf{s}$ are a linear function of $\mathbf{x}$.

$$s_k = \mathbf{v}_k^T \mathbf{x} + a_k$$

(a) Give an expression for $\frac{\partial \ell}{\partial \mathbf{w}}$ and $\frac{\partial \ell}{\partial b_j}$. (Hint: Answer is similar to the previous exercise)

If $j = t$:
$\frac{\partial \ell}{\partial \mathbf{w}_j} = \frac{\partial}{\partial \mathbf{w}_j}(u_t - \log \sum_{k=1}^{C} \exp(u_k))$
$= \frac{\partial}{\partial \mathbf{w}_j}(\mathbf{w}_t^T \mathbf{h} + b_t - \log \sum_{k=1}^{C} \exp(u_k))$

$= h - \frac{\partial}{\partial \mathbf{w}_j}(\log \sum_{k=1}^{C} \exp(\mathbf{w}_k^T \mathbf{h} + b_k))$

$= h - \frac{\exp(\mathbf{w}_j^T \mathbf{h} + b_j) \cdot h}{\sum_{k=1}^{C} \exp(\mathbf{w}_k^T \mathbf{h} + b_k)}$

If $j \neq t$:
$\frac{\partial \ell}{\partial \mathbf{w}_j} = \frac{\partial}{\partial \mathbf{w}_j}(u_t - \log \sum_{k=1}^{C} \exp(u_k))$

$= -\frac{\exp(\mathbf{w}_j^T \mathbf{h} + b_j) \cdot h}{\sum_{k=1}^{C} \exp(\mathbf{w}_k^T \mathbf{h} + b_k)}$

If $j = t$:
$\frac{\partial \ell}{\partial b_j} = \frac{\partial}{\partial b_j}(u_t - \log \sum_{k=1}^{C} \exp(u_k))$
$= \frac{\partial}{\partial b_j}(\mathbf{w}_t^T \mathbf{h} + b_t - \log \sum_{k=1}^{C} \exp(u_k))$

$= \frac{\partial}{\partial b_j}(\mathbf{w}_t^T \mathbf{h} + b_t) - \frac{\partial}{\partial b_j}(\log \sum_{k=1}^{C} \exp(\mathbf{w}_k^T \mathbf{h} + b_k))$

$= 1 - \frac{\exp(\mathbf{w}_j^T \mathbf{h} + b_j)}{\sum_{k=1}^{C} \exp(\mathbf{w}_k^T \mathbf{h} + b_k)}$

If $j \neq t$:
$\frac{\partial \ell}{\partial b_j} = \frac{\partial}{\partial b_j}(u_t - \log \sum_{k=1}^{C} \exp(u_k))$

$= -\frac{\exp(\mathbf{w}_j^T \mathbf{h} + b_j)}{\sum_{k=1}^{C} \exp(\mathbf{w}_k^T \mathbf{h} + b_k)}$

(b) Derive $\frac{\partial \ell}{\partial h_k}$ and show it is equal to $\sum_j \frac{\partial \ell}{\partial u_j} w_{jk}$. (Hint: use $\frac{\partial \ell}{\partial h_k} = \sum_j \frac{\partial \ell}{\partial u_j} \frac{\partial u_j}{\partial h_k}$)

$\frac{\partial \ell}{\partial h_k} = \sum_j \frac{\partial \ell}{\partial u_j} \frac{\partial u_j}{\partial h_k}$ and $\frac{\partial u_j}{\partial h_k} = w_{jk}$
Gives us $\sum_j \frac{\partial \ell}{\partial u_j} w_{jk}$

(c) Derive $\frac{\partial h_k}{\partial s_k}$ and show it is equal to $h_k(1 - h_k)$.

$\frac{\partial h_k}{\partial s_k} = \frac{1}{1+\exp(-s_k)}$
$= (1 + \exp(-s_k))^{-1}$
$= -\frac{1}{(1+\exp(-s_k))^2} \cdot -\exp(-s_k)$
$= \frac{\exp(-s_k)}{(1+\exp(-s_k))^2}$
$= \frac{1}{1+\exp(-s_k)} \cdot \frac{\exp(-s_k)}{1+\exp(-s_k)}$
$= \frac{1}{1+\exp(-s_k)} \cdot \frac{(1+\exp(-s_k))-1}{1+\exp(-s_k)}$
$= \frac{1}{1+\exp(-s_k)} \cdot (\frac{1+\exp(-s_k)}{1+\exp(-s_k)} - \frac{1}{1+\exp(-s_k)})$
$= \frac{1}{1+\exp(-s_k)} \cdot (1 - \frac{1}{1+\exp(-s_k)})$
$= h_k(1 - h_k)$

(d) Derive $\frac{\partial s_k}{\partial \mathbf{v}_k}$ and $\frac{\partial s_k}{\partial a_k}$

$\frac{\partial s_k}{\partial \mathbf{v}_k} = \frac{\partial}{\partial \mathbf{v}_k}(\mathbf{v}_k^T \mathbf{x} + a_k) = x + 0 = x$
$\frac{\partial s_k}{\partial a_k} = \frac{\partial}{\partial a_k}(\mathbf{v}_k^T \mathbf{x} + a_k) = 0 + 1 = 1$

(e) Give the expression for the gradients $\frac{\partial \ell}{\partial \mathbf{v}_k}$ and $\frac{\partial \ell}{\partial a_k}$

$\frac{\partial \ell}{\partial \mathbf{v}_k} = \frac{\partial \ell}{\partial \mathbf{h}_k} \cdot \frac{\partial \mathbf{h}_k}{\partial s_k} \cdot \frac{\partial s_k}{\partial v_k}$
$= (\sum_j \frac{\partial \ell}{\partial u_j} w_j) \cdot h_k(1 - h_k) \cdot x$

$\frac{\partial s_k}{\partial a_k} = \frac{\partial \ell}{\partial \mathbf{h}_k} \cdot \frac{\partial \mathbf{h}_k}{\partial s_k} \cdot \frac{\partial s_k}{\partial a_k}$
$= (\sum_j \frac{\partial \ell}{\partial u_j} w_j) \cdot h_k(1 - h_k)$

3 Open questions

(a) Explain why we need gradients for in neural networks. And why do we not simply set the gradient of the likelihood to zero?

We use gradients to take steps in the right direction using gradient descent to minimize the classification error. If we set the gradient to 0 then it will think it's already minimized so it will not explore further because it thinks it has converged.

(b) Imagine we have a neural network with k layers without non-linearities: $y(\mathbf{x}) = \mathbf{W}_k \dots \mathbf{W}_2 \mathbf{W}_1 \mathbf{x}$. Show or explain that this network cannot learn non-linear functions.

The network can't learn non-linear functions because a combination of linear functions is still linear.

(c) What is an advantage of using mini-batches during the training of a neural network?

Using mini-batches you reduces the variance of the parameter updates, which can lead to a more stable convergence.