Machine Learning Assignment 9

Shivangi Aneja

8-January-2018

Problem 1

The purpose of using basis functions in Neural Networks is as follows:
(1)It transforms the input signal which is not linearly separable into another form which is linearly separable, which can be then feed into the network for processing.
(2)It can also be used for dimensionality reduction in some cases.

Problem 2

The sigmoid function is given as:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

The tan hyperbolic function is given as:

$$tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}} = \frac{e^{2x}-1}{1+e^{2x}} = \frac{e^{2x}}{1+e^{2x}} - \frac{1}{1+e^{2x}} = \frac{e^{2x}}{1+e^{2x}} - \frac{1+e^{2x}-e^{2x}}{1+e^{2x}} = 2\frac{e^{2x}}{1+e^{2x}} - 1 = 2\sigma(2x) - 1$$

$$\boxed{tanh(x) = 2\sigma(2x) - 1}$$

Thus, to transform a layer with sigmoid activation function to tanh activation function:
(a) Half the weights that are input to the neurons to be transformed.
(b) Add 1 to the results $\sum w_i tanh(x_i)$

This output is then fed to the upper layer neurons for the same results as it is an equivalent network.

Problem 3

$$\frac{d(f/g)}{dx} = \frac{gf' - fg'}{g^2}$$

$$\frac{d(tanh)}{dx} = \frac{cosh \times sinh' - sinh \times cosh'}{cosh^2}$$

Now,

$$sinh' = \frac{1}{2}(e^x + e^{-x}) = cosh$$

$$cosh' = \frac{1}{2}(e^x - e^{-x}) = sinh$$

Thus,

$$\frac{d(tanh)}{dx} = \frac{cosh^2 - sinh^2}{cosh^2} = 1 - \frac{sinh^2}{cosh^2} = 1 - tanh^2$$

$$\boxed{\frac{d(tanh(x))}{dx} = 1 - tanh^2(x)}$$

This is a useful property because it is being used in backpropagation while computing the gradient as the gradient does not vanishes.

Problem 4

For logistic Sigmoid Activation function,

$$E(W) = -\sum_{i=1}^{N} \Big(y_i log[f(x_i, W)] + (1 - y_i) log[1 - f(x_i, W)] \Big)$$

$0 \leq f(x_i, W) \leq 1$ and $y_i \in \{0, 1\}$

Using logistic sigmoid for $-1 \leq f(x_i, W) \leq 1$ and $y_i \in \{-1, 1\}$

$$P(y = 1|x, W) = \sigma(f(x, W)) = \frac{1}{1 + e^{-f(x,W)}}$$

$$P(y = -1|x, W) = 1 - \sigma(f(x, W)) = \frac{1}{1 + e^{f(x,W)}}$$

Thus,

$$P(y_i|x_i) = \frac{1}{1 + e^{-y_i f(x_i,W)}}$$

Assuming independence, the likelihood is

$$\prod_{i=1}^{N} \frac{1}{1 + e^{-y_i f(x_i,W)}}$$

The negative log likelihood/Error is
$E(W) = \sum_{i=1}^{N} log(1 + e^{-y_i f(x_i,W)})$

The appropriate choice of activation function in this case will be $\boxed{tanh(x)}$ as for this function , $\boxed{-1 \leq f(x_i, W) \leq 1}$ and this is what we need here.

Problem 5

$E(w) = \frac{1}{m}\sum_{i=1}^{m} \ell(y_i - wx_i) + \lambda||w||^2/2$
where

$$\ell(\eta) = \begin{cases} \frac{1}{2}\eta^2 & |\eta| < 1 \\ |\eta| - \frac{1}{2} & otherwise \end{cases}$$

Or we have,

$$\ell(\eta) = \begin{cases} \frac{1}{2}\eta^2 & -1 < \eta < 1 \\ \eta - \frac{1}{2} & \eta \geq 1 \\ -\eta - \frac{1}{2} & \eta \leq 1 \end{cases}$$

$$\frac{\partial(E(w))}{\partial w} = \frac{1}{m}\sum_{i=1}^{m} \frac{\partial(\ell(y_i - wx_i))}{\partial w} + \lambda||w||$$

Let $\frac{\partial(\ell(y_i - wx_i))}{\partial w} = f_i(w)$

$$f_i(w) = \begin{cases} (wx_i - y_i)x_i & -1 < y_i - wx_i < 1 \\ -x_i & y_i - wx_i \geq 1 \\ x_i & y_i - wx_i \leq -1 \end{cases}$$

$\boxed{\frac{\partial(E(w))}{\partial w} = \frac{1}{m}\sum_{i=1}^{m} f_i(w) + \lambda||w||}$, with $f_i(w)$ given above

Problem 6

When the training error decreases and the validation error increases, it is possible that overfitting might have occurred. Thus , it means that we should stop training.

From the given figure it is clear that:
(A) After 50 iterations the error on validation data starts to increase slightly , and the error on training data decreases rapidly. Also error on test data is also decreasing at this time. So we do stop here as error on validation set increases.
(B) After 100 iterations, we se that error on validation data starts to increase rapidly , but the error on training data decreases slightly and slowly. Also error on test data is also increasing rapidly at this time.

Thus , we stop at updates = 50

Problem 7

$y = log \sum_{i=1}^{N} e^{x_i}$.......(1)

The result of calculating y can be easily overshoot in exponentiation if we calculate this naively.

$y = log \sum_{i=1}^{N} e^{x_i} = log \sum_{i=1}^{N} \frac{C}{C} \times e^{x_i} = log \sum_{i=1}^{N} C \times e^{x_i - log(C)}$

$= log(C) + log \sum_{i=1}^{N} e^{x_i - log(C)}$ ....(2)

Let $log(C) = a$, we have

$y = a + log \sum_{i=1}^{N} e^{x_i - a}$ ....(3)

Setting $a = max_i(x_i)$ results in greatest value to be zero and thus we get a reasonable result

Using (1), (2) and (3) we get,

$$\boxed{y = log \sum_{i=1}^{N} e^{x_i} = a + log \sum_{i=1}^{N} e^{x_i - a}}$$

Problem 8

Exponentiation in the softmax function makes it possible to easily overshoot this number, even for fairly modest-sized inputs.

$$\frac{e^{x_i}}{\sum\limits_{i=1}^{N} e^{x_i}} = \frac{Ce^{x_i}}{\sum\limits_{i=1}^{N} Ce^{x_i}} = \frac{e^{x_i - log(C)}}{\sum\limits_{i=1}^{N} e^{x_i - log(C)}}.....(1)$$

Let $log(C) = a$ ....(2) , we get

Using (1) and (2), we get

$$\boxed{\frac{e^{x_i}}{\sum\limits_{i=1}^{N} e^{x_i}} = \frac{e^{x_i - a}}{\sum\limits_{i=1}^{N} e^{x_i - a}}}$$

Setting $a = max_i(x_i)$ results in greatest value to be zero and thus we get a reasonable result

Problem 9

We have ,

$-[y \times log(\sigma(x)) + (1-y) \times log(1-\sigma(x))]$

$= -[y \times log(\frac{1}{1+e^{-x}}) + (1-y) \times log(\frac{e^{-x}}{1+e^{-x}})]$

$= -[y(-log(1+e^{-x})) + (1-y)(log(e^{-x}) - log(1+e^{-x}))]$

$= -[-y(log(1+e^{-x})) + log(e^{-x}) - log(1+e^{-x}) - y \times log(e^{-x}) + y \times log(1+e^{-x})]$

$= log(1+e^{-x}) - log(e^{-x}) + y \times log(e^{-x})$

$= log(1+e^{-x}) + x - xy$

But we know that $log(1+e^{-x})$ is unstable for as $x \to -\infty$, so we have

$= \boxed{log(1+e^{-abs(x)}) + max(x,0) - xy}$

Hence Proved