CSC413 Deep Learning and Neural Networks

Assignment 1

Yongzhen Huang

January 2020

1 Hard-Coding Networks

1.1 Verify Sort

$$
\begin{aligned}
\mathbf{W}^{(1)} &= \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \\
\mathbf{b}^{(1)} &= \tilde{\mathbf{0}} \\
\mathbf{W}^{(2)} &= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\
b^{(2)} &= -2.5
\end{aligned}
$$

1.2 Perform Sort

1.3 Universal Approximation Theorem

1.3.1

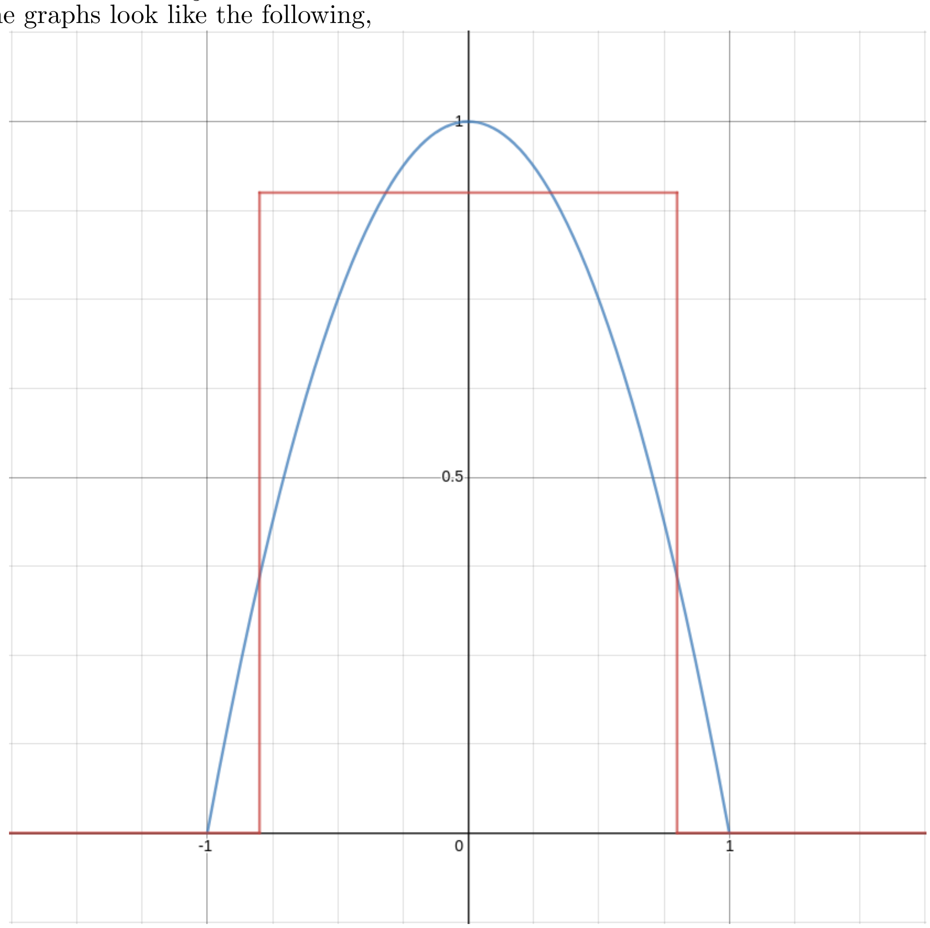
$$
\begin{aligned}
n &= 2 \\
\mathbf{W}^{(0)} &= \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\
\mathbf{b}^{(0)} &= \begin{bmatrix} -a \\ b \end{bmatrix} \\
\mathbf{W}^{(1)} &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
b^{(1)} &= -h
\end{aligned}
$$

1.3.2

$$\int_{-1}^{1}(-x^2+1)dx = (-\frac{1}{3}x^3+x)\Big|_{-1}^{1} = \frac{4}{3}$$

$$\begin{aligned}
||f-\hat{f}_1|| &= \int_{-1}^{1}(-x^2+1-h\cdot I(a<x<b))dx \\
&= \int_{-1}^{a}(-x^2+1)dx + \int_{b}^{1}(-x^2+1)dx + \int_{a}^{b}(-x^2+1-h)dx \\
&\qquad\qquad (-1\le a<b\le 1) \\
&= (-\frac{a^3}{3}+a+\frac{2}{3}) + (\frac{2}{3}+\frac{b^3}{3}-b) + \left|\frac{a^3-b^3}{3}+(1-h)(b-a)\right|
\end{aligned}$$

Want $||f-\hat{f}_1|| < \frac{4}{3}$, a reasonable choice could be $a=-0.8, b=0.8, h=0.9$. The graphs look like the following,



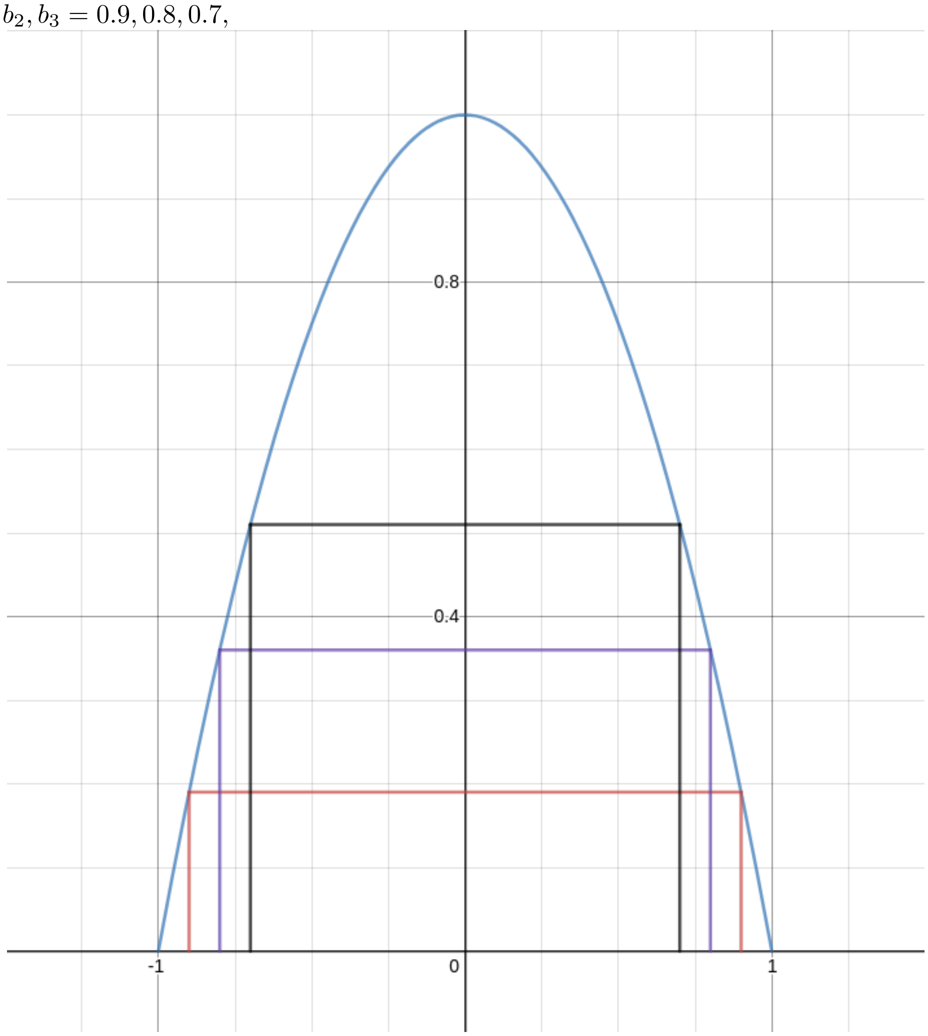
1.3.3

Since f is symmetrix around $x = 0$, $b_i = -a_i$ should be satisified. First, have $\hat{f}_1$ have (h_1, a_1, b_1) such that $h_1 = f(a_1) = f(b_1)$. Then, as we increase a_i and decrease b_i (by the same amount, d_i for instance), we set

$$h_i = f(a_i) - \textstyle\sum_{k=1}^{i-1} h_k$$

Depending on the size of N, we can choose the initial a_1 and the incremental difference (which can be varible) in order to approximate f well.

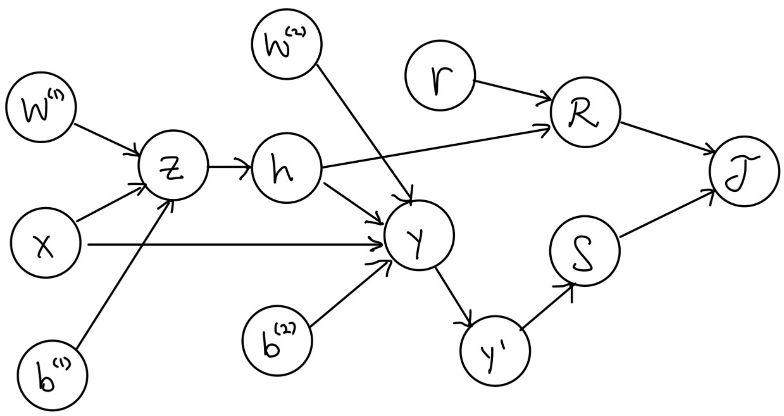
The following plot shows the result of $a_1, a_2, a_3 = -0.9, -0.8, -0.7$ and $b_1, b_2, b_3 = 0.9, 0.8, 0.7$,



2 Backprop

2.1 Computational Graph

2.1.1



2.1.2

$$
\begin{aligned}
\bar{\mathcal{R}} &= 1 \\
\bar{\mathcal{S}} &= 1 \\
\bar{y}'_k &= \bar{\mathcal{S}} \frac{\partial \mathcal{S}}{\partial \bar{y}'_k} \\
&= I(t = k) \\
\bar{\mathbf{y}} &= \bar{y}' \frac{\partial y'}{\partial \mathbf{y}} \\
&= \bar{y}' \circ \texttt{softmax'} \\
\bar{\mathbf{h}} &= \bar{y} \frac{\partial \mathbf{y}}{\partial \mathbf{h}} + \bar{\mathcal{R}} \frac{\partial \mathcal{R}}{\partial \mathbf{h}} \\
&= {\mathbf{W}^{(2)}}^T \bar{y}' + \mathbf{r} \\
\bar{\mathbf{z}} &= \bar{h} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} = \bar{\mathbf{h}} \circ I(\mathbf{v} > 0) \\
\bar{\mathbf{x}} &= \bar{\mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{x}} + \bar{\mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \\
&= {\mathbf{W}^{(1)}}^T \bar{\mathbf{z}} + \bar{\mathbf{y}}
\end{aligned}
$$

2.2 Vector-Jacobian Products (VJPs)

2.2.1

$$J = (vv^T)^T = (vv^T) = \begin{bmatrix}1\\2\\3\end{bmatrix}\begin{bmatrix}1 & 2 & 3\end{bmatrix} = \begin{bmatrix}1 & 2 & 3\\2 & 4 & 6\\3 & 6 & 9\end{bmatrix}$$

2.2.2

Both the time and memory cost are $\mathcal{O}(n^2)$.

2.2.3

$\mathbf{z} = J^T\mathbf{y} = \mathbf{v}\mathbf{v}^T\mathbf{y} = \mathbf{v}(\mathbf{v}^T\mathbf{y})$. First, compute $\alpha = v^T\mathbf{y}$ which can be achieved in time and space $\mathcal{O}(n)$. Then, compute $\mathbf{z} = \alpha\mathbf{v}$ which again uses time and space $\mathcal{O}(n)$. For the example,

$$\begin{aligned}\alpha &= \mathbf{v}^T\mathbf{y}\\ &= 6\\ \mathbf{z} = \alpha\mathbf{v} &= \begin{bmatrix}6\\12\\18\end{bmatrix}\end{aligned}$$

3 Linear Regression

3.1 Deriving the Gradient

3.2 Underparametrized Model

3.2.1

$$\begin{aligned}\mathcal{L} &= \frac{1}{n}(X\hat{\mathbf{w}} - \mathbf{t})^2\\ &= \frac{1}{n}(X\hat{\mathbf{w}} - \mathbf{t})^T(X\hat{\mathbf{w}} - \mathbf{t})\\ &= \frac{1}{n}\left(\hat{\mathbf{w}}X^TX\hat{\mathbf{w}} + \mathbf{t}^T\mathbf{t} - 2\mathbf{t}^TX\hat{\mathbf{w}}\right)\\ \frac{\partial\mathcal{L}}{\partial\hat{\mathbf{w}}} &= \frac{2}{n}\left(X^TX\hat{\mathbf{w}} - X^T\mathbf{t}\right)\end{aligned}$$

3.2.2

$$
\begin{aligned}
0 &= \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{w}}} \\
&= X^T X \hat{\mathbf{w}} - X^T \mathbf{t} \\
X^T \mathbf{t} &= X^T X \hat{\mathbf{w}} \\
\hat{\mathbf{w}} &= (X^T X)^{-1} X^T \mathbf{t}
\end{aligned}
$$

Since $X^T X$ is invertible, the solution is unique.

3.3 Overparametrized Model: 2D Example

3.3.1

Since $d < n$, we have that all $span(X) = R^d$. So, let $x \in R^d$, then,

$$x = \sum_{i=1}^{n} \alpha_i x_i, x_i \in X$$

Note that if we plug in $\hat{\mathbf{w}}$ into the loss function $\mathcal{L}$, we would get 0. Then,

$$
\begin{aligned}
&(\mathbf{w}^{*T} x - \hat{\mathbf{w}}^T x) \\
=&\sum_{i=1}^{N} \alpha_i (\mathbf{w}^{*T} - \hat{\mathbf{w}}^T) x_i \\
=&0
\end{aligned}
$$

3.3.2

From 3.2.1, we are trying to solve

$$
\begin{aligned}
X^T t &= X^T X \hat{\mathbf{w}} \\
\begin{bmatrix} 4 \\ 2 \end{bmatrix} &= \begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \end{bmatrix} \hat{\mathbf{w}} \\
\begin{bmatrix} 4 \\ 2 \end{bmatrix} &= \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} \hat{\mathbf{w}}
\end{aligned}
$$

From here, we have that any $\hat{\mathbf{w}}$ satisfying the line

$$\hat{\mathbf{w}}_2 = 2 - 2\hat{\mathbf{w}}_1$$

will satisfy this equation and thus there are infinitely many solutions.

3.3.3

With $\hat{\mathbf{w}}(0) = 0$, we get that the direction of gradient is

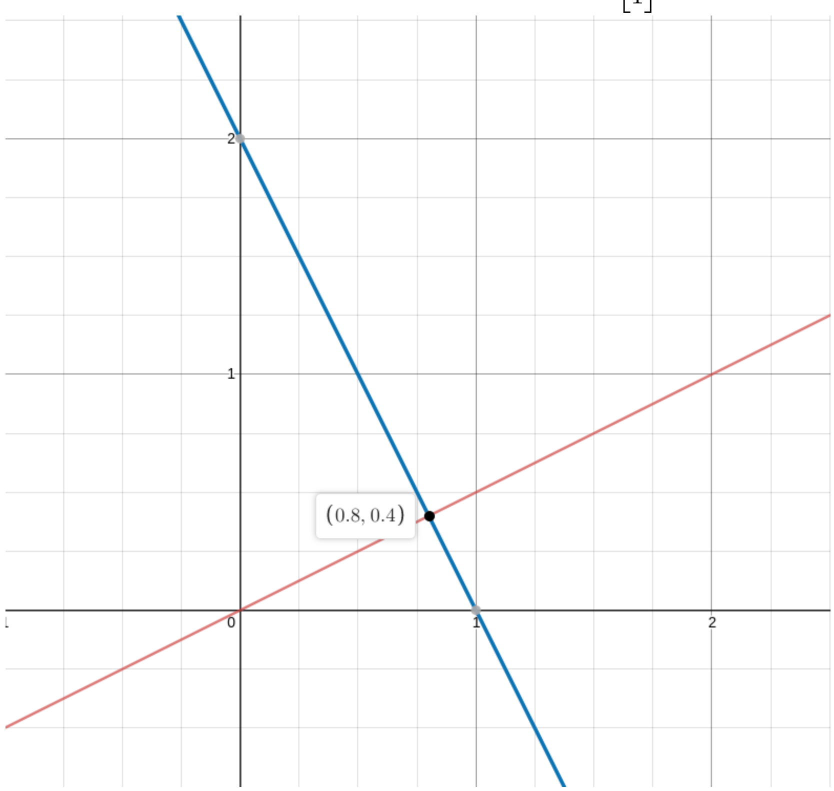
$$\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{w}}} = -2X^T\mathbf{t} = \begin{bmatrix} -8 \\ -4 \end{bmatrix} \implies \frac{1}{\sqrt{5}} \begin{bmatrix} -2 \\ -1 \end{bmatrix}$$

.

From hereon, notice that this direction is perpendicular to the line we found above. Futhermore, if we plug the updated $\hat{\mathbf{w}}$ along this direction and evaluate the gradient again, we are still travelling along this line. This is because when we do $y = \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} x$, we still get that $y_2 = 2y_1$, which is exactly the direction of the line. So, overall, the gradient will only travel along this line, with direction $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and y-intercept of 0. Now, gradient descent will go towards the solution that minimizes the loss, which is the intersection between this line $y = \frac{1}{2}x$ and the line $\hat{\mathbf{w}}$ satisfies, $y = 2 - 2x$. So, for the solution, we have the intersection

$$\frac{1}{2}x = 2 - 2x \implies x = \frac{4}{5}, y = \frac{2}{5}$$

So, the gradient descent should find the solution $\hat{\mathbf{w}} = \frac{2}{5} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$.



From the figure, we can see that these two lines are perpendicular to each other. Note that the Euclidean norm of a vector can be seen as the distance from the point to the origin. The optimal solution, $\mathbf{w}^*$, as found above is highlighted in the figure and it can be seen that for any other point $p_1 \neq \mathbf{w}^*$ on the red line (the one $\hat{\mathbf{w}}$ resides on), $\mathbf{w}^*, p_1$, and the origin form a right triangle. By Pythagorean Theorem,

$$||p_1||^2 = ||\mathbf{w}^*||^2 + ||\mathbf{w}^* - p_1||^2$$

So, since $p_1 \neq \mathbf{w}^*$, we have that the Euclidean norm of $\mathbf{w}^*$ is strictly less than that of p_1 and therefore the gradient descent finds the solution with smallest Euclidean norm.

3.4 Overparametrized Model: General Case

3.4.1

The gradient descent seems to find the solution that satisfies the following, 1. The direction it takes is spanned by the rows of X. 2. The solution it finds is the intersection between the span of the rows of X (i.e. the directions) and the space that $\hat{\mathbf{w}}$ resides in. 3. The Euclidean norm of the solution is minimized amongest all (since we start $\mathbf{w}$ at $\vec{0}$).

So, we can re-write our minimization problem to the following,

$$\min_{\hat{\mathbf{w}}} \hat{\mathbf{w}}^2 \text{ s.t. } X\hat{\mathbf{w}} = \mathbf{t}$$

Now, we can use Lagrange multiplier and yield the following,

$$\begin{aligned}
\mathcal{L} &= \hat{\mathbf{w}}^2 - \lambda^T(\mathbf{t} - X\hat{\mathbf{w}}) \\
\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{w}}} &= 2\hat{\mathbf{w}} - X^T\lambda \\
0 &:= 2\hat{\mathbf{w}} - X^T\lambda \\
X^T\lambda &= 2\hat{\mathbf{w}} \\
XX^T\lambda &= 2X\hat{\mathbf{w}} \\
\lambda &= 2(XX^T)^{-1}X\hat{\mathbf{w}} \\
\lambda &= 2(XX^T)^{-1}\mathbf{t}
\end{aligned}$$

Now, we plug in the value of λ into $X^T\lambda = 2\hat{\mathbf{w}}$ and get

$$\hat{\mathbf{w}} = X^T(XX^T)^{-1}\mathbf{t}$$

3.4.2

$$
\begin{aligned}
\hat{\mathbf{w}} &= X^T(XX^T)^{-1}t \\
(\hat{\mathbf{w}} - \hat{\mathbf{w}}_1)^T\hat{\mathbf{w}} &= (X^T(XX^T)^{-1}t - \hat{\mathbf{w}}_1)^T X^T(XX^T)^{-1}t \\
&= t^T(XX^T)^{-T}XX^T(XX^T)^{-1}t - \hat{\mathbf{w}}_1^T X^T(XX^T)^{-1}t \\
&= t^T(XX^T)^{-1}t - \hat{\mathbf{w}}_1^T X^T(XX^T)^{-1}t \\
&= (t - X\hat{\mathbf{w}}_1)^T(XX^T)^{-1} \\
&= 0 && \text{(since } \hat{\mathbf{w}}_1 \text{ is zero-loss)}
\end{aligned}
$$

This value shows that the vectors $\hat{\mathbf{w}}$ and $\mathbf{u} = \hat{\mathbf{w}} - \hat{\mathbf{w}}_1$ are normal to each other. Similar to the figure in 3.3.3, $\hat{\mathbf{w}}$ is our optimal value and implicitly the vector from the origin to this point in space. On the other hand, $\hat{\mathbf{u}}$ is some vector in the space of gradient direction (analogous to a vector on the blue line in 3.3.3). Since $\hat{\mathbf{w}}$ and $\hat{\mathbf{u}}$ are perpendicular to each other, we can show similarly to 3.3.3 by Pythagorean Theorem that this solution $\hat{\mathbf{w}}$ has the smallest Euclidean norm. In particular, consider the following proof.

Suppose $\hat{\mathbf{w}}$ does not have the smallest Euclidean distance so there exists w^* which is the optimal solution so that $||\mathbf{w}^*||^2 < ||\hat{\mathbf{w}}||^2$ for the sake of contradiction. Then, consider three lines: l_1 connecting $\vec{\mathbf{0}}$ to $\hat{\mathbf{w}}$, l_2 connecting $\vec{\mathbf{0}}$ to $\mathbf{w}^*$, and l_3 connecting $\hat{\mathbf{w}}$ to $\mathbf{w}^*$. It is clear that these three points form a triangle. And $\texttt{length}(l_1) = ||\hat{\mathbf{w}}||^2, \texttt{length}(l_2) = ||\mathbf{w}^*||^2$. Furthermore, since $(\mathbf{w}^* - \hat{\mathbf{w}})\hat{\mathbf{w}} = 0$, l_1 and l_3 are normal to each other and l_2 is the diagonal. By Pythagorean Theorem, l_2 is the longest, and therefore contradiction. Thus, $||\mathbf{w}^*||^2 > ||\hat{\mathbf{w}}||^2$.

3.5 Benefit of Overparametrization

3.5.1

Overparametrization doesn't seem to always lead to larger test error, such can be seen for $n = 70$.

```

def fit_poly(X, d,):
  X_expand = poly_expand(X, d=d, poly_type = poly_type)
  if d < n:
    W = linalg.inv(X_expand.T@X_expand)@X_expand.T@t
  else:
    W = X_expand.T@linalg.inv(X_expand@X_expand.T)@t
  return W
```

Listing 1: Linear Regression