Homework 3

1. Weight Decay

$$\mathcal{J}(\hat{w}) = \frac{1}{2n}\|X\hat{w} - t\|_2^2 + \frac{\lambda}{2}\hat{w}^T\hat{w}.$$

1.1. Underparametrized Model

if $d \le n$, X^TX is invertible.

$$\underset{d\times 1}{\frac{\partial \mathcal{J}(\hat{w})}{\partial \hat{w}}} = \frac{1}{2n}(2)\underset{d\times n}{X^T}\underset{n\times 1}{(X\hat{w} - t)} + \frac{\lambda}{2}(2)\underset{d\times 1}{\hat{w}}$$

$$= \frac{1}{n}(X^TX\hat{w} - X^Tt) + \lambda\hat{w} = 0.$$

$$\frac{1}{n}X^TX\hat{w} - \frac{1}{n}X^Tt + \lambda\hat{w} = 0$$

$$(X^TX + \lambda nI)\hat{w} = X^Tt$$

we can show that $X^TX + \lambda nI$ is invertible.
X^TX is positive semidefinite, λnI is definite given that $\lambda \ne 0$, so $X^TX + \lambda nI$ is positive definite.
All of its eigenvalues are strictly positive, so 0 is not an eigenvalue. Now looking at

$$(X^TX + \lambda nI)v = 0 \cdot v = 0$$

if this holds for some $v \ne 0$ then 0 is an eigenvalue, contradiction!
therefore, $\hat{w} = (X^TX + \lambda nI)^{-1}X^Tt$.

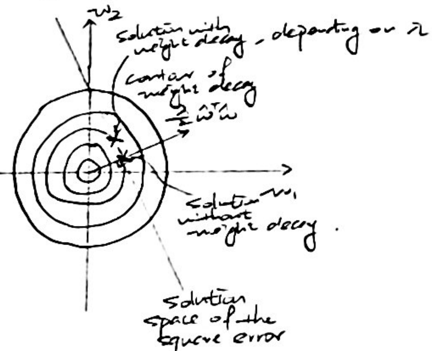
1.2 Overparametrized Model

$\boxed{1.2.1}$ Warmup: Visualizing Weight Decay

from HW1, we have that empirical risk minimizers satisfy

$$X\hat{w}^* = t$$

$$2w_1^* + w_2^* = 2$$



1.2.2 Gradient Descent and Weight Decay

if $d > n$, XX^T is invertible.

$$\nabla w(0) = \frac{1}{n}(0 - X^Tt) = \frac{-1}{n}X^Tt$$

with 0 weight initialization, the gradient direction is $X^Tt \in \mathbb{R}^{d\times 1}$.
it cannot be the same as HW1 3.4.1 since the gradient changes along the way.

$\boxed{1.3}$ Adaptive optimizer and Weight Decay

AdaGrad with weight decay:

$$G_{i,t} = (1-\gamma)G_{i,t-1} + \gamma(\nabla_{w_{i,t}}\mathcal{I}(w_{i,t}))^2$$

$$w_{i,t+1} = w_{i,t} - \frac{\eta}{\sqrt{G_{i,t}} + \varepsilon}\nabla_{w_{i,t}}\mathcal{I}(w_{i,t})$$

look at the 2D toy example first, with $w_0 = \begin{bmatrix}0\\0\end{bmatrix}$.

$$\nabla_{w_0}\mathcal{I}(w_0) = \begin{bmatrix}2\\1\end{bmatrix}(-2) = \begin{bmatrix}-4\\-2\end{bmatrix}.$$

$$G_{1,0} = (1-\gamma)0 + \gamma(-4)^2 = 16\gamma$$

$$G_{2,0} = (1-\gamma)0 + \gamma(-2)^2 = 4\gamma$$

$$w_{1,1} = 0 - \frac{\eta}{\sqrt{16\gamma}+\varepsilon}(-4) = \frac{4\eta}{\sqrt{16\gamma}+\varepsilon} \approx \frac{\eta}{\sqrt{\gamma}}$$

$$w_{2,1} = 0 - \frac{\eta}{\sqrt{4\gamma}+\varepsilon}(-2) = \frac{2\eta}{\sqrt{4\gamma}+\varepsilon} \approx \frac{\eta}{\sqrt{\gamma}}.$$

No, the weights still leaves the span of X, since the weight update is not along $\begin{bmatrix}2\\1\end{bmatrix}$.

2. Ensembles and Bias-Variance Decomposition.

$\boxed{2.1.1}$ Weight Average or Prediction Average.

Suppose there're n models,
$h_1(x;D_1), h_2(x;D_2), \ldots, h_n(x;D_n)$.
using prediction averaging, we have that

$$z_{avg} = \frac{1}{n}(z_1 + z_2 + \ldots + z_n)$$

$$= \frac{1}{n}(X\hat{w}_1 + X\hat{w}_2 + \ldots + X\hat{w}_n + b_1 + \ldots + b_n)$$

where $\hat{w}_1, \ldots, \hat{w}_n$ are learned from $D_1, \ldots, D_n$ respectively.
using weight averaging, we have that

$$z_{avg}' = X\frac{1}{n}(\hat{w}_1 + \hat{w}_2 + \ldots + \hat{w}_n) + \frac{1}{n}(b_1 + \ldots + b_n).$$

$z_{avg} = z_{avg}'$, so the ensemble model using prediction averaging and weight averaging with linear activation are the same, so their generalization errors should be the same, too.

2.1.2

No, $z_{avg} \ne z_{avg}'$

2.2 Bagging – Uncorrelated Models.

$$\bar{h}(x;D) = \frac{1}{k}\sum_{i=1}^{k} h(x;D_i)$$

2.2.1

we need to show that

$$E[\bar{h}(x;D)|x] = E[h(x;D)|x].$$

$$LHS = E\left[\frac{1}{k}\sum_{i=1}^{k} h(x;D_i)|x\right]$$
$$= \frac{1}{k} E\left[\sum_{i=1}^{k} h_i(x;D_i)|x\right]$$
$$= \frac{1}{k} \sum_{i=1}^{k} E[h_i(x;D_i)|x]$$
$$= \frac{1}{k} k\, E[h(x;D_i)|x]$$
$$= E[h(x;D)|x] = RHS.$$

$\boxed{2.2.2}$

$$Var(\bar{h}(x;D)|x)$$
$$= E[|\bar{h}(x;D) - E[\bar{h}(x;D)|x]|^2]$$
$$= E\left[\left|\frac{1}{k}\sum_{i=1}^{k} h(x;D_i) - E\left[\frac{1}{k}\sum_{i=1}^{k} h(x;D_i)|x\right]\right|^2\right]$$
$$= \frac{1}{k^2} E\left[\left|\sum_{i=1}^{k} h(x;D_i) - E\left[\sum_{i=1}^{k} h(x;D_i)|x\right]\right|^2\right]$$
$$= \frac{1}{k^2} E\left[\sum_{i=1}^{k} |h(x;D_i) - E[h(x;D_i)|x]|^2\right]$$
$$= \frac{1}{k^2}\cdot k \cdot \sigma^2$$
$$= \frac{\sigma^2}{k}.$$

2.3 Bagging – General case.

$$\rho = \frac{cov(h(x;D_j), h(x;D_k))}{\sigma_j \sigma_k}, \quad \forall j \neq k.$$

correlation,

$\boxed{2.3.1}$ Bias under correlation

Due to linearity of expectation

$E[aX] = aE[X]$

$E[X+Y] = E[X] + E[Y]$

for any $a \in \mathbb{R}$ and random variables X and Y.

(2.2.1)

the same reasoning as before still applies.

that $E\left[\frac{1}{k}\sum_{i=1}^{k} h(x;D_i)|x\right] = \frac{1}{k}\sum_{i=1}^{k} E[h(x;D_i)|x]$

$= E[h(x;D)|x]$.

So the bias term doesn't change.

2.3.2 Variance under Correlation.

$$\text{variance} = \left(\rho + \frac{1-\rho}{k}\right)\sigma^2$$

$\boxed{2.3.3}$ Intuitions on Bagging

looking at the variance equation, increasing k will lower the variance.

when $\rho = 0$, we recover the variance when ensemble members are not correlated, i.e. we get $\frac{1}{k}\sigma^2$ back.

when $\rho = 1$, we get σ^2 variance, ensemble doesn't decrease variance when ensemble members are fully correlated. this makes sense since in this case we don't get any new information.

3. Generalization and Dropout.

3.1.1 Regress from x_1.

$$\hat{y} = w_1 x_1$$
$$J = E[(Y - w_1 X_1)^2]$$
$$= E[Y^2] - 2w_1 E[X_1 Y] + w_1^2 E[X_1^2].$$
$$\frac{\partial J}{\partial w_1} = -2E[X_1 Y] + 2w_1 E[X_1^2] = 0$$
$$E[X_1 Y] = w_1 E[X_1^2].$$
$$\underbrace{\cancel{E[X_1]}}_{0} E[Y] + cov(X_1, Y) = w_1 E[X_1^2]$$
$$w_1 E[X_1^2] = cov(X_1, Y)$$
$$w_1 E[(X_1 - 0)^2] = cov(X_1, Y)$$
$$w_1\, cov(X_1, X_1) = cov(X_1, Y)$$
$$w_1 = \frac{cov(X_1, Y)}{cov(X_1, X_1)}$$
$$= \frac{cov(X_1, X_1 + N(0,\sigma^2))}{\sigma^2}$$
$$= \frac{cov(X_1, X_1) + \cancel{cov(X_1, N(0,\sigma^2))}}{\sigma^2}$$
$$= 1.$$

$\boxed{3.1.2}$ Regress from x_2

$$\hat{y} = w_2 x_2.$$
$$J = E[(Y - w_2 X_2)^2]$$
$$= E[Y^2] - 2w_2 E[Y X_2] + w_2^2 E[X_2^2]$$
$$\frac{\partial J}{\partial w_2} = -2E[Y X_2] + 2w_2 E[X_2^2]$$
$$w_2 = \frac{E[Y X_2]}{E[X_2^2]} = \frac{\cancel{E[Y]E[X_2]} + cov(X_2, Y)}{1}$$
$$= cov(X_2, Y)$$
$$= cov(X_1 + N(0,\sigma^2) + N(0,1),\ X_1 + N(0,\sigma^2))$$

6 terms.

$$= cov(X_1, X_1) + \cancel{cov(X_1, N(0,\sigma^2))} + cov(N(0,\sigma^2), X_1)$$
$$+ cov(N(0,\sigma^2), N(0,\sigma^2)) + cov(N(0,1), X_1)$$
$$+ \cancel{cov(N(0,1), N(0,\sigma^2))}$$
$$= \sigma^2 + \sigma^2$$
$$= 2\sigma^2$$

$\boxed{3.1.3}$ Regress from (x_1, x_2).

$\hat{y} = w_1 x_1 + w_2 x_2$.

$J = E[(Y - w_1 x_1 - w_2 x_2)^2]$

$= E[Y^2 - 2w_1 x_1 Y + w_1^2 x_1^2 - 2w_2 x_2 Y + 2w_1 w_2 x_1 x_2 + w_2^2 x_2^2]$.

$= E[Y^2] + w_1^2 E[x_1^2] + w_2^2 E[x_2^2] - 2w_1 E[x_1 Y] - 2w_2 E[x_2 Y] + 2w_1 w_2 E[x_1 x_2]$.

$= cov(Y, Y) + w_1^2 cov(x_1, x_1) + w_2^2 cov(x_2, x_2) - 2w_1 cov(x_1, Y) - 2w_2 cov(x_2, Y) + 2w_1 w_2 cov(x_1, x_2)$.

$\frac{\partial J}{\partial w_1} = 2w_1 cov(x_1, x_1) - 2 cov(x_1, Y) + 2w_2 cov(x_1, x_2)$

$= 2w_1 \sigma^2 - 2\sigma^2 + 2w_2 \sigma^2$

$= (2w_1 + 2w_2 - 2)\sigma^2 = 0$, $\quad 2w_1 = 2 - 2w_2$, $w_1 = 1 - w_2$

$\frac{\partial J}{\partial w_2} = 2w_2 cov(x_2, x_2) - 2cov(x_2, Y) + 2w_1 cov(x_1, x_2)$.

$= 2w_2(2\sigma^2 + 1) - 4\sigma^2 + 2w_1\sigma^2$

$= 4w_2\sigma^2 + 2w_2 - 4\sigma^2 + 2w_1\sigma^2$

$4w_2\sigma^2 + 2w_2 - 4\sigma^2 + 2(1 - w_2)\sigma^2 = 0$

$4w_2\sigma^2 + 2w_2 - 4\sigma^2 + 2\sigma^2 - 2w_2\sigma^2 = 0$

$4w_2\sigma^2 + 2w_2 - 2\sigma^2 - 2w_2\sigma^2 = 0$.

$(4\sigma^2 + 2 - 2\sigma^2)w_2 = 2\sigma^2$.

$\cancel{2}(\sigma^2 + 1)w_2 = \cancel{2}\sigma^2$

$w_2 = \frac{\sigma^2}{\sigma^2 + 1}$, $w_1 = 1 - \frac{\sigma^2}{\sigma^2 + 1} = \frac{1}{\sigma^2 + 1}$.

At test time, $\hat{y} = w_1 x_1 + w_2 x_2$

$= \frac{1}{\sigma^2 + 1} x_1 + \frac{\sigma^2}{\sigma^2 + 1} x_2$ is a weighted sum of x_1 and x_2

Suppose σ^2 gets smaller (bigger) during the test time,
w_1 will increase (decrease) and w_2 will decrease (increase).
So x_1 has more (less) effect on the output.
and x_2 has less (more) effect on the output

In the extreme case, if $\sigma^2 = 0$ for training, then
$w_1 = 1$ $w_2 = 0$.
$\hat{y} = x_1$.
if $\sigma^2 \neq 0$ for test, then prediction is off, worse generalize.

$\boxed{3.3}$. Effect on Dropout

$E_m[J] = (E_m[\hat{y}] - t)^2 + \frac{1}{2}\sum_j Var[x_j] w_j^2$

$= (E_m[2m_1 w_1 x_1 + 2m_2 w_2 x_2] - y)^2 + (w_1^2\sigma^2 + w_2^2(\sigma^2 + 1))$

$= (w_1 x_1 + w_2 x_2 - y)^2 + w_1^2\sigma^2 + w_2^2(\sigma^2 + 1)$

$= E[w_1^2 x_1^2 + w_2^2 x_2^2 + y^2 + 2w_1 w_2 x_1 x_2 - 2w_1 x_1 y - 2w_2 x_2 y] + w_1^2\sigma^2 + w_2^2\sigma^2 + w_2^2$

$= w_1^2\sigma^2 + w_2^2(2\sigma^2 + 1) + 2\sigma^2 + 2w_1 w_2 \sigma^2 - 2w_1\sigma^2 - 2w_2(2\sigma^2) + w_1^2\sigma^2 + w_2^2\sigma^2 + w_2^2$

$= 2w_1^2\sigma^2 + 4w_2^2\sigma^2 + 2w_2^2 + 2\sigma^2 + 2w_1 w_2 \sigma^2 - 2w_1\sigma^2 - 4w_2\sigma^2$

$\frac{\partial E_m[J]}{\partial w_1} = 4w_1\sigma^2 + 2w_2\sigma^2 - 2\sigma^2 = 0$

$\frac{\partial E_m[J]}{\partial w_2} = 8w_2\sigma^2 + 4w_2 + 2w_1\sigma^2 - 4\sigma^2 = 0$

$w_1 = \frac{2\sigma^2 + 2}{7\sigma^2 + 4}$

$w_2 = \frac{3\sigma^2}{7\sigma^2 + 4}$

this is better because the weights don't change a lot with σ^2, since both the numerator and denominator are second order, this will result in more generalizability.

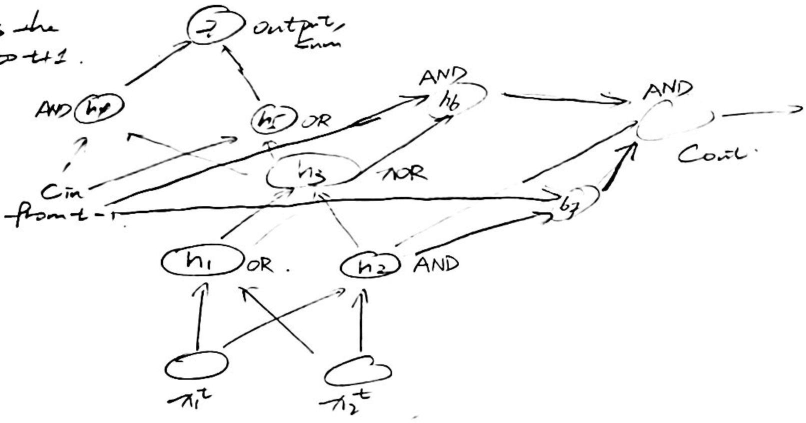
4. Hard-coding Recurrent Neural Networks.

truth table: this is the memory from $t-1$

x_1^t	x_2^t	C_{in}^t	Sum^t	C_{out}^t
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

this is the input to $t+1$.

this is the output.



$$Sum^t = XOR(x_1^t, x_2^t, C_{in}^t)$$
$$= NOT[XOR(XOR(x_1^t, x_2^t), C_{in}^t)]$$

$$C_{out}^t = OR[AND(x_1^t, x_2^t), AND(x_1^t, C_{in}^t), AND(x_2^t, C_{in}^t)]$$
$$= C_{in}^{t+1}$$