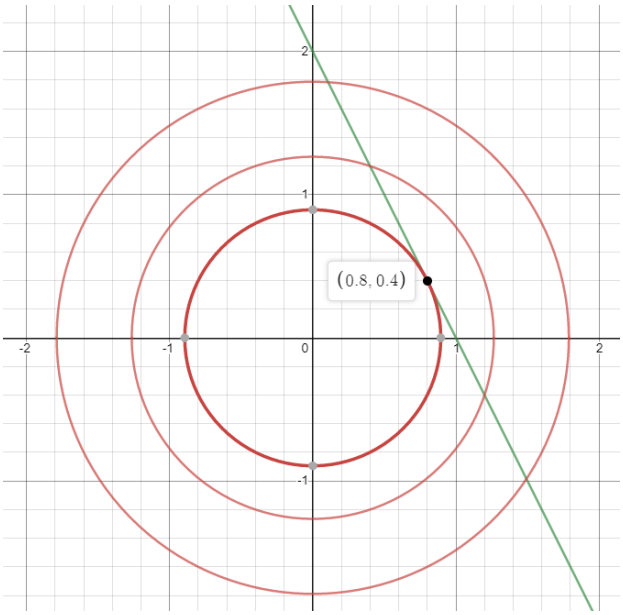
1.2.1



It is shown that the solutions with and without weight decay is the same which is (0.8, 0.4)

1.3

Weight decay does help AdaGrad to converge to a solution in the row space because weight decay penalizes weights with larger norms. Since AdaGrad finds an optimal solution, then it must lie within the solution space. Therefore, weight decay will help push the solution to go to the point on the solution space that has minimum norm, thus keeping the solution within the row space.

2.1

We can represent weight average as: $h(x, D) = \left(\frac{1}{k}\sum_{i=1}^{k} w_i\right) x$ and prediction average as: $h(x, D) = \frac{1}{k}\sum_{i=1}^{k} y_i$. However, note that they are equivalent: $\left(\frac{1}{k}\sum_{i=1}^{k} w_i\right) x = \frac{1}{k}\sum_{i=1}^{k} w_i\, x_i = \frac{1}{k}\sum_{i=1}^{k} y_i$.
Therefore, the ensemble of linear models using weight average or prediction average gives the same expected generalization error.

2.2.2

$$\mathbb{E}\left[\left|\bar{h}(x;D) - \mathbb{E}[\bar{h}(x;D)|x]\right|^2\right] = \mathbb{E}\left[\left|\frac{1}{K}\sum_{i=1}^{K} h(x;D_i) - \frac{1}{K}\sum_{i=1}^{K}\mathbb{E}[h(x;D_i)|x]\right|^2\right]$$

$$= \frac{1}{K^2}\,\mathbb{E}\left[\left|\sum_{i=1}^{K}\left(h(x;D_i) - \mathbb{E}[h(x;D_i)|x]\right)\right|^2\right] = \frac{1}{K^2}\sum_{i=1}^{K}\mathbb{E}\left[\left|h(x;D_i) - \mathbb{E}[h(x;D_i|x]\right|^2\right)$$

$$= \frac{1}{K^2}\left(K\,\mathbb{E}\left[\left|h(x;D) - \mathbb{E}[h(x;D)|x]\right|^2\right]\right) = \frac{1}{K}\,\sigma^2$$

2.3.1

The bias term does not change because expectation of averages is still the same even if the models are correlated.

3.1.2

Let $x_1 \leftarrow \varepsilon_1$, $\varepsilon_1 \sim N(0, \sigma^2)$ $\quad E[\varepsilon_1^2] = V[\varepsilon_1] + E[\varepsilon_1]^2 = \sigma^2$

$y \leftarrow x_1 + \varepsilon_2$, $\varepsilon_2 \sim N(0, \sigma^2)$ $\quad E[\varepsilon_2^2] = \sigma^2$

$x_2 \leftarrow y + \varepsilon_3$, $\varepsilon_3 \sim N(0, 1)$ $\quad E[\varepsilon_3^2] = 1$

given $\hat{y} = w_1 x_2$

$$J = E[(y-\hat{y})^2] = E[(y - w_1 x_2)^2] = E[(y - w_1(y+\varepsilon_3))^2]$$

$$= E[((1-w_1)(\varepsilon_1+\varepsilon_2) - w_1\varepsilon_3)^2]$$

dropping the cross terms after expansion due to independence:

$$= w_1^2 E[\varepsilon_1^2] - 2w_1 E[\varepsilon_1^2] + E[\varepsilon_1^2] + w_1^2 E[\varepsilon_2^2] - 2w_1 E[\varepsilon_2^2] + E[\varepsilon_2^2] + w_1^2 E[\varepsilon_3^2]$$

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

$$\frac{\partial J}{\partial w_1} = 2w_1\sigma^2 - 2\sigma^2 + 2w_1\sigma^2 - 2\sigma^2 + 2w_1 = 0$$

$$2w_1\sigma^2 - 2\sigma^2 + w_1 = 0$$

$$w_1 = \frac{2\sigma^2}{2\sigma^2+1}$$

3.1.3

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

$$J = E[(y-\hat{y})^2] = E[(y - (w_1 x_1 + w_2 x_2))^2] = E[((\varepsilon_1+\varepsilon_2) - (w_1\varepsilon_1 + w_2(\varepsilon_1+\varepsilon_2+\varepsilon_3)))^2]$$

dropping the cross terms after expansion due to independence:

$$= 2w_1w_2 E[\varepsilon_1^2] + w_1^2 E[\varepsilon_1^2] - 2w_1 E[\varepsilon_1^2] + w_2^2 E[\varepsilon_1^2] - 2w_2 E[\varepsilon_2^2]$$

$$+ w_2^2 E[\varepsilon_2^2] - 2w_2 E[\varepsilon_2^2] + w_2^2 E[\varepsilon_3^2] + E[\varepsilon_2^2]$$

$$= 2w_1w_2\sigma^2 + w_1^2\sigma^2 - 2w_1\sigma^2 + w_2^2\sigma^2 - 2w_2\sigma^2 + w_2^2\sigma^2 - 2w_2\sigma^2 + w_2^2 + \sigma^2$$

$$\frac{\partial J}{\partial w_1} = 2w_1\sigma^2 + 2w_1\sigma^2 - 2\sigma^2 = 0 \quad \Rightarrow \quad w_1 = \frac{2\sigma^2 - 2w_2\sigma^2}{2\sigma^2} \quad (1)$$

$$\frac{\partial J}{\partial w_2} = 2w_1\sigma^2 + 2w_2\sigma^2 - 2\sigma^2 + 2w_2\sigma^2 - 2\sigma^2 + 2w_2 = 0 \quad (2)$$

Sub (1) into (2): ~~$2\sigma^2 - 2w_2\sigma^2 + 2w_2\sigma^2 - 2\sigma^2$~~ $+ 2w_2\sigma^2 - 2\sigma^2 + 2w_2 = 0$

$$w_2 = \frac{2\sigma^2}{2\sigma^2+2} = \frac{\sigma^2}{\sigma^2+1}$$

sub back to (1) we get:

$$w_1 = \frac{1}{\sigma^2+1}$$

We can see that if σ becomes smaller during test time, then w_1 will become close to 1 and w_2 will become close to 0. Therefore, this maximum likelihood will likely not generalize well if σ changes.

3.3

given $\hat{y} = 2(m_1 w_1 x_1 + m_2 w_2 x_2)$

$$J = E[(y-\hat{y})^2] = E[(y - 2(m_1 w_1 x_1 + m_2 w_2 x_2))^2] = E\left[\left((\varepsilon_1+\varepsilon_2) - 2(m_1 w_1 \varepsilon_1 + m_2 w_2 (\varepsilon_1+\varepsilon_2+\varepsilon_3))\right)^2\right]$$

dropping the cross terms after expansion due to independence:

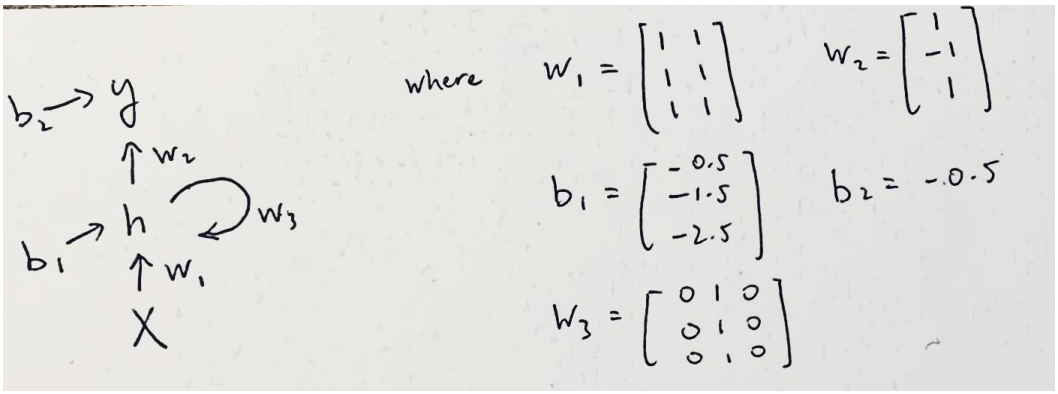
$$= 8 w_1 w_2 \left(\tfrac{1}{2}\right)^2 E[\varepsilon_1^2] + 4 w_1^2 \left(\tfrac{1}{2}\right)^2 E[\varepsilon_1^2] - 4 w_1 \left(\tfrac{1}{2}\right) E[\varepsilon_1^2] + 4 w_2^2 \left(\tfrac{1}{2}\right)^2 E[\varepsilon_1^2]$$

$$- 4 w_1 \left(\tfrac{1}{2}\right) E[\varepsilon_1^2] + 4 w_2^2 \left(\tfrac{1}{2}\right)^2 E[\varepsilon_2^2] + 4 w_2^2 \left(\tfrac{1}{2}\right)^2 E[\varepsilon_3^2] + E[\varepsilon_1^2] + E[\varepsilon_2^2]$$

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

We can see that this results to the same equation as 3.1.2. Therefore, in this model, dropout does not help since it got cancelled out. Thus, it will generalize poorly as well.

4



Note that the above RNN uses threshold function as activation function.

The above RNN will be able to detect when the inputs are either (1) at least 1 input is nonzero (2) when both inputs are 1 and there is no carry (3) when both input is 1 and there is carry from previous time step.

Then it will output accordingly: (1) output 1 and no carry for next time step (2) output 0 and carry 1 to next time step (3) output 1 and carry 1 to next time step.