1. for convex f with L_2 proximity term

$$w^{(t+1)} = \arg\min_w \; f(w^{(t)}) + \langle w - w^{(t)}, \nabla f(w^{(t)}) \rangle + \frac{\lambda}{2} \| w - w^{(t)} \|^2$$

derive w.r.t. w

$$\frac{d}{dw} = \nabla f(w^{(t)}) + \lambda (w - w^{(t)}) = 0$$

$$\Rightarrow \nabla f(w^{(t)}) + \lambda w - \lambda w^{(t)} = 0$$

$$\Rightarrow \lambda w - \lambda w^{(t)} + \nabla f(w^{(t)}) = 0$$

$$\Rightarrow \lambda w = \lambda w^{(t)} - \nabla f(w^{(t)})$$

$$\Rightarrow w^{(t+1)} = w^{(t)} - \frac{1}{\lambda} \nabla f(w^{(t)})$$

$$\Rightarrow w^{*} = w^{(t)} - \frac{1}{\lambda} \nabla f(w^{(t)})$$

This gives us the same form as the gradient descent update rule. Note that $\frac{1}{\lambda} = \eta$. This tells us that gradient descent minimizes the approximation at each step. The relationship tells us that as learning rate increases, regularization must decrease.

2. show $\sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle \le \frac{\|w^*\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|v_t\|^2$

Take LHS: $\sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle$

$= \sum_{t=1}^{T} \langle w^{(t)}, v_t \rangle - \sum_{t=1}^{T} \langle w^*, v_t \rangle$

$= \underbrace{\sum_{t=1}^{T} \langle w^{(t)}, v_t \rangle}_{①} - \underbrace{\langle w^*, \sum_{t=1}^{T} v_t \rangle}_{②}$

aside: consider update eq'n $w^{(t+1)} = w^{(t)} - \eta v_t$, $w^{(1)} = 0$

$w^2 = w^1 - \eta v_1 = -\eta v_1$

$w^3 = w^2 - \eta v_2 = -\eta v_1 - \eta v_2$

$\vdots$

$w^{(t+1)} = w^{(t)} - \eta v_t = -\eta v_1 - \cdots - \eta v_t$

$= -\eta \sum_{i=1}^{t} v_i$ ③

Take ① and ③:

$\sum_{t=1}^{T} \langle w^{(t)}, v_t \rangle = \sum_{t=1}^{T} \langle -\eta \sum_{i=1}^{t-1} v_i, v_t \rangle$

$= -\eta \sum_{t=1}^{T} \langle \sum_{i=1}^{t-1} v_i, v_t \rangle$

$= -\eta \left(\frac{\langle \sum_{t=1}^{T} v_t, \sum_{t=1}^{T} v_t \rangle}{2} - \frac{\sum_{t=1}^{T} \|v_t\|^2}{2} \right)$

$= -\frac{\eta}{2} \| \sum_{t=1}^{T} v_t \|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|v_t\|^2$

$= \frac{-1}{2\eta} \|w^{(t+1)}\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|v_t\|^2$

Take ② and ③:

since $w^{(t+1)} = -\eta \sum_{t=1}^{T} v_t \Rightarrow \sum_{t=1}^{T} v_t = \frac{-1}{\eta} w^{(t+1)}$

$\langle w^*, \sum_{t=1}^{T} v_t \rangle = \langle w^*, \frac{-1}{\eta} w^{(t+1)} \rangle$

∴ LHS $= \frac{-1}{2\eta} \|w^{(t+1)}\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|v_t\|^2 + \frac{1}{\eta} \langle w^*, w^{(t+1)} \rangle$

RHS $= \frac{\|w^*\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|v_t\|^2$ terms cancel out

We need to show $\underbrace{\frac{-1}{2\eta} \|w^{(t+1)}\|^2}_{④} + \underbrace{\frac{1}{\eta} \langle w^*, w^{(t+1)} \rangle}_{⑤} \le \underbrace{\frac{\|w^*\|^2}{2\eta}}_{⑥}$

since ④ is the negative product of a positive term ($\|w^{(t+1)}\|^2$), ④ ≤ 0

take ⑤: $\frac{1}{\eta} \langle w^*, w^{(t+1)} \rangle = \frac{1}{2\eta} \langle w^*, w^{(t+1)} \rangle + \frac{1}{2\eta} \langle w^*, w^{(t+1)} \rangle$

Note $\langle w^*, w^{(t+1)} \rangle \leq 0$ and $\frac{1}{2n}\langle w^*, w^{(t+1)} \rangle \leq \frac{1}{2n}\|w^*\|^2$

$\therefore -\frac{1}{2n}\|w^{(t+1)}\|^2 + \frac{1}{2n}\langle w^*, w^{(t+1)} \rangle \leq 0$

Thus our inequality is

$$\underbrace{-\frac{1}{2n}\|w^{(t+1)}\|^2 + \frac{1}{2n}\langle w^*, w^{(t+1)} \rangle}_{\text{this term} \leq 0} + \frac{1}{2n}\langle w^*, w^{(t+1)} \rangle \leq \frac{\|w^*\|^2}{2n}$$

this term ≤ 0

and the rest of the inequality is already proven

$\therefore$ adding a value ≤ 0 holds the inequality ▨

3. For $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w^{(t)}$ we get $f(\bar{w}) - f(w^*) \le \frac{1}{T}\sum_{t=1}^{T} \langle w^{(t)} - w^*, \nabla f(w^{(t)}) \rangle$

by the definition of convexity:

$$f(w^*) \ge f(w^{(t)}) + \langle w^* - w^{(t)}, \nabla f(w^{(t)}) \rangle$$

$$\Rightarrow f(w^*) - f(w^{(t)}) \ge \langle w^* - w^{(t)}, \nabla f(w^{(t)}) \rangle$$

$$\Rightarrow f(w^{(t)}) - f(w^*) \le \langle w^{(t)} - w^*, \nabla f(w^{(t)}) \rangle$$

$$\Rightarrow \frac{1}{T}\sum_{t=1}^{T} [f(w^{(t)})] - f(w^*) \le \frac{1}{T}\sum_{t=1}^{T} \langle w^{(t)} - w^*, \nabla f(w^{(t)}) \rangle$$

Since f is convex, we know $f(\bar{w}) = f(\frac{1}{T}\sum_{t=1}^{T} w^{(t)}) \le \frac{1}{T}\sum_{t=1}^{T} f(w^{(t)})$

$$\Rightarrow f(\bar{w}) - f(w^*) \le \frac{1}{T}\sum_{t=1}^{T} \langle w^{(t)} - w^*, \nabla f(w^{(t)}) \rangle$$

From q2: $\sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle \le \frac{\|w^*\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \|v_t\|^2$

$$\Rightarrow f(\bar{w}) - f(w^*) \le \frac{1}{T}\left[\frac{\|w^*\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \|\nabla f(w^{(t)})\|^2\right]$$

$$\Rightarrow f(\bar{w}) - f(w^*) \le \frac{\|w^*\|^2}{2T\eta} + \frac{\eta}{2T}\sum_{t=1}^{T} \|\nabla f(w^{(t)})\|^2$$

upper bound $\|w^*\|$ with B

$\|\nabla f(w^{(t)})$ with ρ

$$\Rightarrow f(\bar{w}) - f(w^*) \le \frac{B^2}{2T\eta} + \frac{\eta}{2T}\sum_{t=1}^{T} \rho^2$$

$$\Rightarrow f(\bar{w}) - f(w^*) \le \frac{B^2}{2T\eta} + \frac{\eta T \rho^2}{2T}$$

sub in $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$ and reduce

$$f(\bar{w}) - f(w^*) \le \frac{B^2}{2T\sqrt{\frac{B^2}{\rho^2 T}}} + \frac{\sqrt{\frac{B}{\rho^2 T}}\,\rho^2}{2}$$

$$\vdots$$

$$f(\bar{w}) - f(w^*) \le \frac{B\rho}{\sqrt{T}}$$

$\therefore$ convergence rate of w^* is $O(\frac{1}{\sqrt{T}})$ and upper bound for $f(\bar{w}) - f(w^*) \propto \frac{1}{\sqrt{T}}$

4. a) $g(x) = \underset{w}{\operatorname{argmin}} \left[f(w^{(t)}) + \langle w - w^{(t)}, \nabla f(w^{(t)}) \rangle + \frac{1}{2} (w - w^{(t)})^T H (w - w^{(t)}) \right]$

take the derivative and set $= 0$

$$\nabla g(x) = \nabla f(w^{(t)}) + H(w - w^{(t)}) = 0$$

$$\Rightarrow w - w^{(t)} = -H^{-1} \nabla f(w^{(t)})$$

$$\Rightarrow w = w^{(t)} - H^{-1} \nabla f(w^{(t)})$$

↑ this is the form of Newton's update method

H^{-1}, the inverse of the Hessian, shapes the gradient, giving emphasis on different directions. We can use the direction given by the Hessian to make the next iteration step.

4.b) $x \in \mathbb{R}^{50} \rightarrow h^{(1)} \in \mathbb{R}^{50} \rightarrow h^{(2)} \in \mathbb{R}^{50} \rightarrow s \in \mathbb{R}^{10}$

$50 \times 50 + 50 \times 50 + 50 \times 50$ parameters in the MLP

$5500^2 * 10$ dimensional Hessian

4.c) $\nabla_w(w+\Delta w) = \nabla_w(w) + H\Delta w + O(\|\Delta w\|^2)$

let $\Delta w = rv$

$\Rightarrow H\Delta w = \nabla_w(w+\Delta w) - \nabla_w(w) - O(\|\Delta w\|^2)$

$\Rightarrow Hrv = \nabla_w(w+rv) - \nabla_w(w) - O(\|rv\|^2)$

$\Rightarrow Hv = \frac{1}{r}\left[\nabla_w(w+rv) - \nabla_w(w) + O(r^2)\right]$

$Hv = \frac{1}{r}\left[\nabla_w(w+rv) - \nabla_w(w)\right] + O(r)$

4.d) $R_v\{f(w)\} = \frac{d}{dr} f(w+rv) = f'(w+rv) * (w+rv)dr = vf'(w+rv)$

proof of linearity:

$$R_v\{cf(w)\} = \frac{d}{dr} cf(w+rv) = c\frac{d}{dr} f(w+rv) = cvf'(w+rv)$$

$$cR_v\{f(w)\} = cvf'(w+rv)$$

$\therefore R_v\{cf(w)\} = cR_v\{f(w)\}$ ■

Proof of Product Rule:

$$R_v\{f(w)g(w)\} = \frac{d}{dr}\left(f(w+rv)g(w+rv)\right)$$

$$= f(w+rv)\frac{dg(w+rv)}{dr} + g(w+rv)\frac{df(w+rv)}{dr}$$

$$= vf(w+rv)g'(w+rv) + vf'(w+rv)g(w+rv)$$

$$R_v\{f(w)\}g(w) + f(w)R_v\{g(w)\} = vf'(w+rv)g(w+rv) + vf(w+rv)g'(w+rv)$$

$\therefore R_v\{f(w)g(w)\} = R_v\{g(w)\}g(w) + f(w)R_v\{g(w)\}$ ■

4.e) $Hv = \frac{1}{r}\left[\nabla_w(w+rv) - \nabla_w(w)\right] + O(r)$

$R_v\{f(w)\} = \frac{d}{dr}(w+rv)\Big|_{r=0}$

take the limit of Hv as $r \to 0$:

$$Hv = \lim_{r\to 0} \frac{\nabla_w(w+rv) - \nabla_w(w)}{r}$$

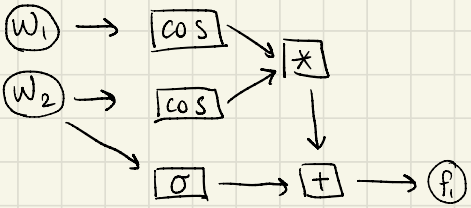
$$= \frac{d}{dr}\nabla_w(w+rv)\Big|_{r=0}$$

$\Rightarrow Hv = R_v\{\nabla_w(w)\}$ ▨

4.f) We can apply the Hv calculation in backpropagation of MLPs to derive R{backprop}. This allows us to avoid calculating the Hessian, and compute the gradient efficiently. With auto differentiation, we can use the implicit matrix products by using the forward-over-reverse method.

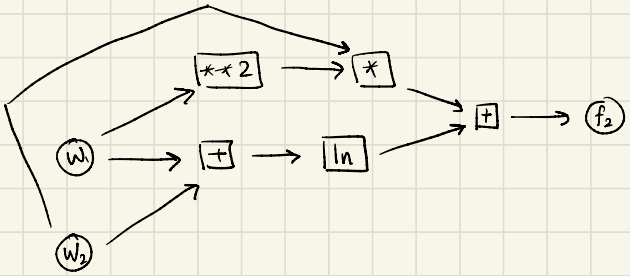
5. a)

$f_1(w_1, w_2) = \cos(w_1)\cos(w_2) + \sigma(w_2)$ $\quad w = (1, 2)$



$f_1(w) = \cos(1)\cos(2) + \sigma(2) = 0.656$

$f_2(w_1, w_2) = \ln(w_1 + w_2) + w_1^2 w_2$



$f_2(w) = \ln(1+2) + (1^2)(2) = 3.097$

5.b) $\frac{df}{dw} = J = \begin{bmatrix} \frac{df_1}{dw_1} & \frac{df_1}{dw_2} \\ \frac{df_2}{dw_1} & \frac{df_2}{dw_2} \end{bmatrix}$ $\Delta w = 0.01$

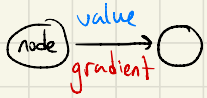
$\frac{df_1}{dw_1} = \frac{f_1(1.01, 2) - f_1(1,2)}{0.01} = 0.351$ $\frac{df_1}{dw_2} = \frac{f_1(1, 2.01) - f_1(1,2)}{0.01} = -0.386$

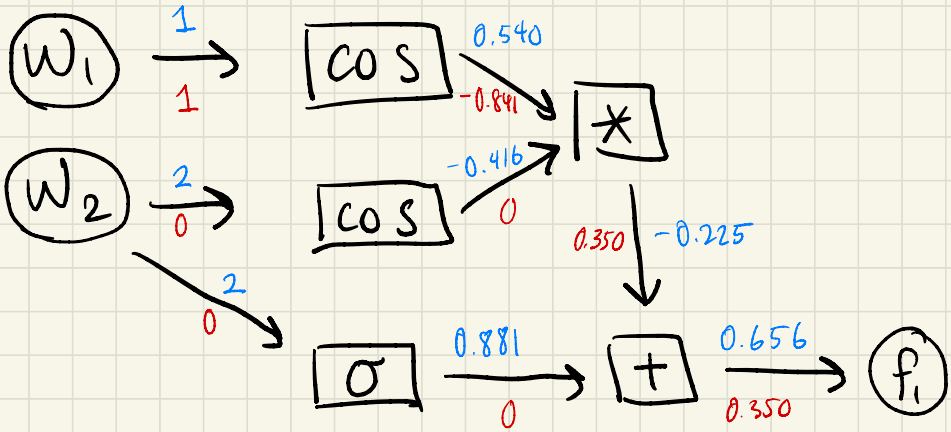
$\frac{df_2}{dw_1} = \frac{f_2(1.01, 2) - f_2(1,2)}{0.01} = 4.353$ $\frac{df_2}{dw_2} = \frac{f_2(1, 2.01) - f_2(1,2)}{0.01} = 1.333$

$J = \begin{bmatrix} 0.351 & -0.386 \\ 4.353 & 1.333 \end{bmatrix}$

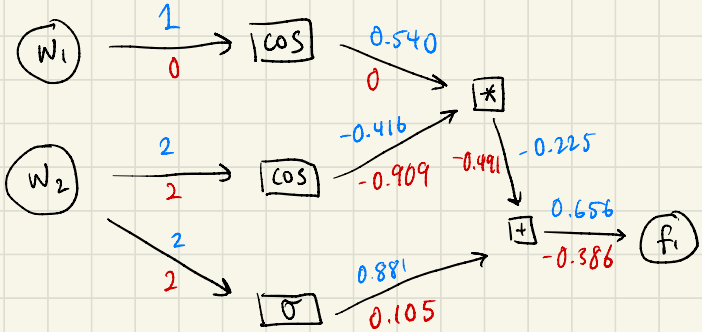
5. c)

Forward Mode AD $\frac{d}{dw_1} f_1(w_1, w_2)$

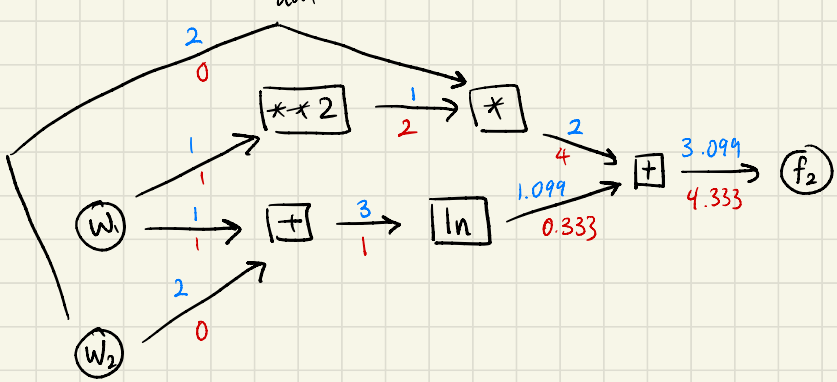




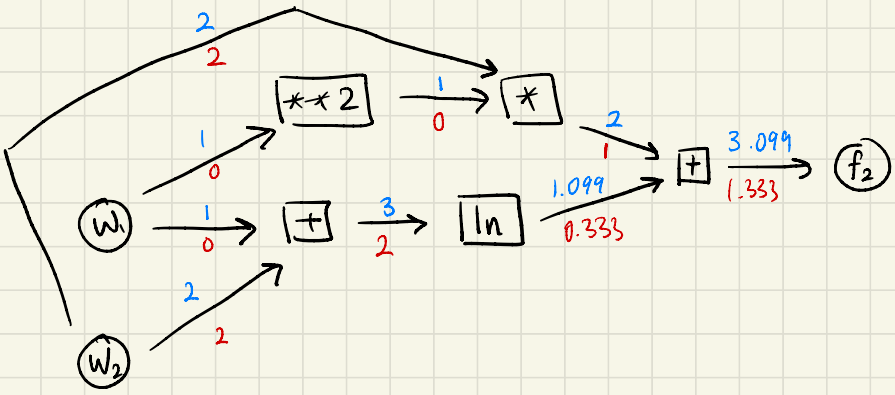
Forward Mode AD $\frac{d}{dw_2} f_1(w_1, w_2)$



Forward Mode AD $\frac{d}{dw_1} f_2(w_1, w_2)$

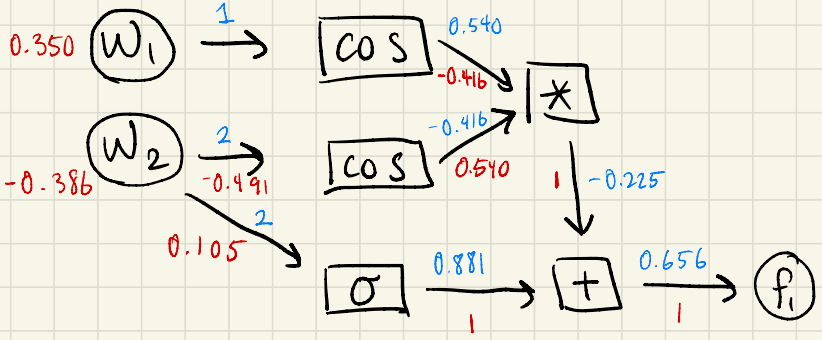


Forward Mode AD $\frac{d}{dw_2} f_2(w_1, w_2)$

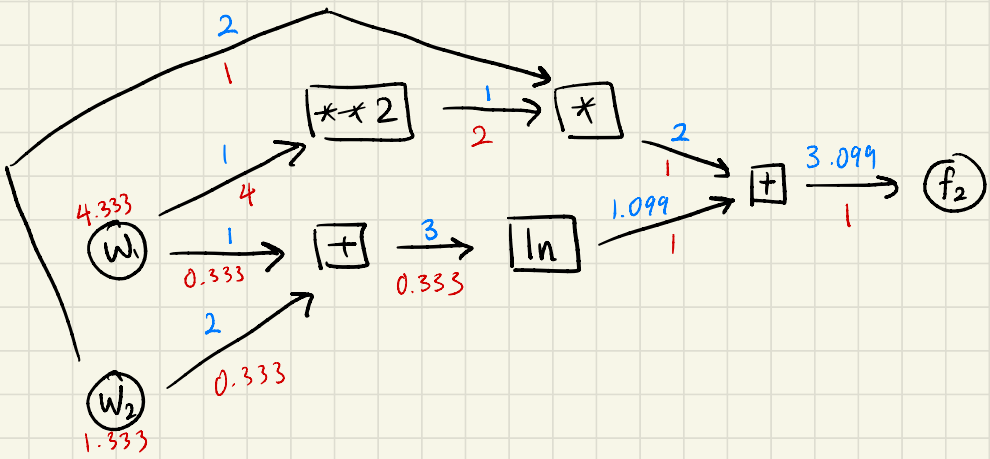


$$J = \begin{bmatrix} 0.550 & -0.386 \\ 0.333 & 1.333 \end{bmatrix}$$

5. d) Backward Mode AD $\frac{d}{dw} f_1(w_1, w_2)$



Backward Mode AD $\frac{d}{dw} f_2(w_1, w_2)$



$$J = \begin{bmatrix} 0.350 & -0.386 \\ 4.333 & 1.333 \end{bmatrix}$$

5. e) This is my 5th - 6th time doing AD by hand and I'm thinking this is revenge of the machines

6. a) S is the circular right shift operator.
Let S^* be the circular left shift operator

$$Sx = \begin{bmatrix} 0 & & & 1 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} x_{n-1} \\ x_0 \\ \vdots \\ x_{n-2} \end{bmatrix}$$

$$S^*x = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & \\ 1 & & & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \\ x_0 \end{bmatrix}$$

From this it is clear that S^* is the transpose of S, thus also the adjoint
Note both are circulant matrices
The actions of S and S^* can be expressed in terms of vector indices and modular arithmetic:

$$(Sx)_k = x_{k-1}, \quad (S^*x)_k = x_{k+1} \qquad \text{where e.g. } (Sx)_0 = x_{0-1} \equiv_n x_{n-1}$$

The ij^{th} entry of S can be expressed using the Kronecker delta:

$$(S)_{ij} = \delta_{i-j-1}$$

Let C_a be any Circulant matrix where the entries of C_a are denote a_i

$$(S C_a)_{ij} = \sum_\ell S_{i\ell} (C_a)_{\ell j} = \sum_\ell \delta_{i-\ell-1} a_{\ell-j} = \sum_\ell \delta_{(i-1)-\ell} a_{\ell-j} = a_{i-1-j}$$

$$(C_a S)_{ij} = \sum_\ell (C_a)_{i\ell} S_{\ell j} = \sum_\ell a_{i-\ell} \delta_{\ell-j-1} = \sum_\ell a_{i-\ell} \delta_{\ell-(j+1)} = a_{i-j-1} = a_{i-1-j}$$

We can see that $SC_a = C_a S \ \forall \ C_a$ ▨

6. b) Prove: Matrix M is circulant iff it commutes with the circular shift operator S.

$\Rightarrow$: we proved the forward direction in part a)

$\Leftarrow$: if M commutes with S, then M is circulant

$$SM = MS$$

We use the same Kronecker delta and index expression as in part a

$$(SM)_{ij} = \sum_\ell S_{i\ell} (M)_{\ell j} = \sum_\ell \delta_{i-\ell,1} (M)_{\ell j} = (M)_{i-1,j}$$

$$(MS)_{ij} = \sum_\ell (M)_{i\ell} S_{\ell j} = \sum_\ell (M)_{i\ell} \delta_{\ell-j,1} = (M)_{i,j+1}$$

Since $SM = MS$:

$$\underbrace{(M)_{i-1,j} = (M)_{i,j+1}}_{①} \iff \underbrace{(M)_{i-1,j-1} = (M)_{ij}}_{②} \iff \underbrace{(M)_{ij} = (M)_{i+1,j+1}}_{③}$$

Eq'n ③ implies that for any k, $(M)_{ij} = (M)_{i+k,j+k}$

$\Rightarrow$ Entries of M are constant along diagonals

take 1st column of M as $m_i = (M)_{i,0}$

$\Rightarrow (M)_{ij} = (M)_{i-j,j-j} = (M)_{i-j,0} = m_{i-j}$

$\therefore$ all entries of M are obtained from the first column by circular shifts ■

6. c) We can use circulant matrices to store weights in deep learning. Since circulant matrices have reduced number of independent parameters, we can do fast multiplication using it.

7. a) Any situation where digital hardware is reading continuous information, there could be errors in the readings. Such as reconstructing sounds from samples

7. b) $\underset{z}{\operatorname{argmin}} \; E[L(z,y)]$

$L1 : \sum_{i=1}^{n} |z - y_i|$

The minimum is found at median $\{\vec{y}\}$

7.c) $L2 = \sum_{i=1}^{n} (z - y_i)^2$

The minimum is found at the mean of $\vec{y}$: $E_y\{\vec{y}\}$

7. d) $D' = \{ (x_i, y_i \pm \epsilon) \mid (x_i, y_i) \in D \;,\; \epsilon \sim N(0, \sigma) \}$

Mean would stay the same

$E_y\{y\}$

7. e) $\underset{\theta}{\operatorname{argmin}} \; E_{(x,y)}[L(f_\theta(x), y)]$

$E(X|Y) = \sum x \times P(X = x | y)$

Law of joint distribution

$P(A \cap B) = P(A) \cdot P(B|A)$

$\Rightarrow \underset{\theta}{\operatorname{argmin}} \; E_{(x,y)}[L(f_\theta(x), y)] = \underset{\theta}{\operatorname{argmin}} \; E_x[E_{y|x}[L(f_\theta(x), y)]]$

7.f) $\underset{\theta}{\operatorname{argmin}} \sum_i L[f_\theta(\hat{x}_i, \hat{y}_i)]$ where both $\hat{x}_i$ and $\hat{y}_i$ are drawn from a corrupted distribution.

On the condition that the unobserved clean target $\hat{y}_i$ s.t. $E\{\hat{y}_i \mid \hat{x}_i\} = y_i$

From parts a - d, we know $z = E_y\{y\}$ holds no matter the distribution from which y's are drawn and the optimal network parameters of $\underset{\theta}{\operatorname{argmin}} E_x\{E_{y|x}\{L(f_\theta(x), y)\}$ remain unchanged

7. g) Given infinite data, the soln to $\underset{\theta}{\operatorname{argmin}} \sum_i L\left[f_\theta(\hat{x}_i, \hat{y}_i)\right]$ is the same as $\underset{\theta}{\operatorname{argmin}} \sum_i L\left(f_\theta(\hat{x}_i), y_i\right)$, which is the original given mapping between corrupt and clean observations

8. No.

Let $w^{(1)} = 0$

$$f(w) = \frac{1}{2}(w-2)^2 + \frac{1}{2}(w+1)^2$$

$$= \frac{1}{2}(w^2 - 4w + 4) + \frac{1}{2}(w^2 + 2w + 1)$$

$$= w^2 - w + \frac{5}{2}$$

$$f(w^{(1)}) = f(0) = \frac{5}{2}$$

Pick second term for next gradient descent

$$w^{(2)} = w^{(1)} - \eta(w^{(1)} + 1)$$

$$= -\eta$$

$$f(w^{(2)}) = f(-\eta)$$

$$= \eta^2 + \eta + \frac{5}{2}$$

$\Rightarrow f(w^{(2)}) > f(w^{(1)})$

$\Rightarrow$ SGD not guaranteed to decrease overall loss function in every iteration

Key contributions
The paper's key findings "problematize the traditional view of generalization by showing it is incapable of distinguishing between different neural networks that have radically different generalization performance." The results of the experiments can be summarized in that deep neural networks easily fit random labels. The paper further shows that these results rule out many of the possible explanations for generalization performance of state-of-the-art neural networks. They provide a notion of the effective capacity of machine learning models.

Strengths
The paper presents thorough experiments and results that show that the training process for multiple standard architectures is unaffected by whether there is a relationship between instances and the class labels. It shows that their findings imply deep neural networks are able to fit perfectly any set of labels. They show that the effective capacity of several successful architectures is large enough to "shatter the training data."

Weaknesses
Even though the paper provides experimentation, results, and theory on the traditional view of generalization, it did not reach a solution or approach to tell apart models that generalize well. The paper provides an experimental framework for the effective capacity of machine learning models, but it was unable to provide a scientific way/method of measuring such effective capacity. Ultimately, it does not provide a concrete answer to the question it proposes in the beginning: "what distinguishes neural networks that generalize well from those that don't?"

My Takeaways

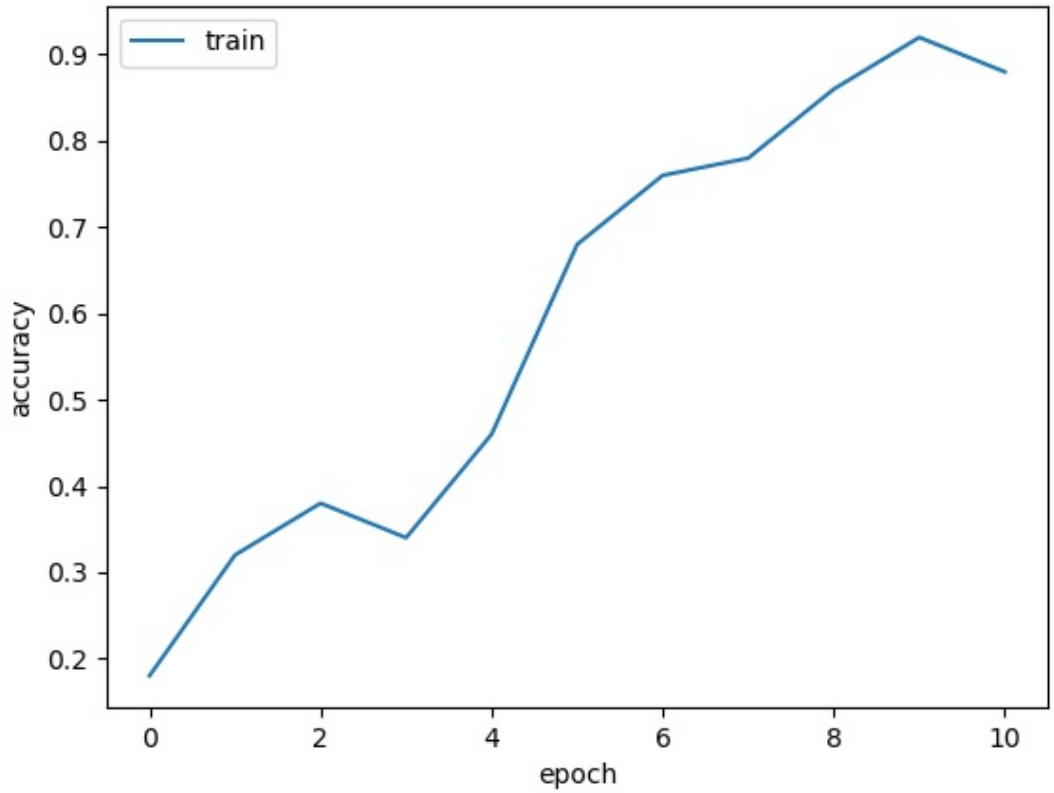
The paper's experiments and findings that explicit forms of regularization do not adequately explain the generalization error of neural nets resonated with some of my prior experience when looking for hyperparameters. I found that regularization did not have too much of an impact on my small model. The paper also makes me think how the difference between models that generalize vs ones that do not (our incomplete understanding of it) could be related to the fact that neural nets are uninterpretable by us. If we could accomplish one or the other, we may be able to learn more about the other.

Collaborators:
Machine learning theory book, pytoch docs, research papers given

Assignment 2 Writeup

Name: Bojun Yang
GT Email: byang301@gatech.edu
GT ID: byang301

Part-1 ConvNet


train
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
accuracy
0
2
4
6
8
10
epoch

My EvalAI Model

Model description: My model was inspired by architectures for cifar10 I found in papers. All of my layers are followed by a GELU activation. It is a sequential neural net that begins with 4 sets of 2 conv layers (6 total conv) with increasing hidden layers from 128 to 512. I found that the depth of the model was more effective in increasing accuracy than width so I increased the depth instead of the width of the conv blocks. I apply max pooling, bach norm, and dropout(0.1) to each set. Then I simply have 3 fully connected layers that classify the flattened output from the conv layers. The kernel size for my conv layers was 3, stride = 1, and padding = 1.

Hyperparameter choice: I kept the hyperparameters the same as the vanilla cnn except I increased learning from 0.0001 to 0.01. I did this because I saw my accuracy was generally increasing, but never reached equilibrium. I decreased the size of the kernel from 7 to 3 because the images were already 32x32 and so a smaller kernel would be able to better filter out details from an already small image. After figuring out more optimal hyperparameters, I increased the number of epochs to get more training in and gradually increased the number of hidden nodes of each layer.

EvalAI acc: My final accuracy on EvalAi was 0.91, under the username byang301.

Data Wrangling

What's your result of training with regular CE loss on imbalanced CIFAR-10?

Fill in your per-class accuracy in the table

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
CE Loss	.833	.8060	.2900	.1770	.0110	.0020	0	0	0	0

What's your result of training with CB-Focal loss on imbalanced CIFAR-10?

Tune the hyper-parameter beta and fill in your per-class accuracy in the table

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
beta=0.9999	.5720	.1500	.1700	.3430	0	0	0	0	0	0
beta=0.1	1	0	0	0	0	0	0	0	0	0

Put your results of CE loss and CB-Focal Loss(best) together:

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
CE Loss	.833	.8060	.2900	.1770	.0110	.0020	0	0	0	0
CB-Focal	.5720	.1500	.1700	.3430	0	0	0	0	0	0

I observed that CB focal loss was able to achieve higher accuracies on classes with less labels in the data.