ECE 046211 - Technion - Deep Learning

HW2 - Multilayer NNs and Convolutional NNs

Part 1 - Theory

- You can choose whether to answser these straight in the notebook (Markdown + Latex) or use another editor (Word, LyX, Latex, Overleaf...) and submit an additional PDF file, **but no handwritten submissions**.
- You can attach additional figures (drawings, graphs,...) in a separate PDF file, just make sure to refer to them in your answers.
- $\LaTeX$ Cheat-Sheet (to write equations)
  - Another Cheat-Sheet

Question 1 -Generalization in A Teacher-Student Setup

Recall from lecture 4 the Bayes Risk $\overline{\mathcal{R}}(w)$:

$$\overline{\mathcal{R}}(w) \triangleq \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma_\epsilon^2 I),w_{true}\sim\mathcal{N}(0,\frac{\sigma_w^2}{d}I)}\left[\mathcal{R}\right],$$

where,

$$\mathcal{R}(w_\mu) = ||w_\mu - w_{true}||^2 = ||(H_\mu^{-1}H - I)w_{true} + H_\mu^{-1}X^T\epsilon||^2$$

Prove:

$$\overline{\mathcal{R}}(w_\mu) = \sum_{i=1}^{d}\frac{(\sigma_w^2/d)\mu^2 + \sigma_\epsilon^2\lambda_i}{(\lambda_i+\mu)^2}$$

Hints:

- $\mathbb{E}\left[\epsilon^T X H_\mu^{-1}H_\mu^{-1}X^T\epsilon\right] = \sum_{i,j}^{N}\mathbb{E}[\epsilon_i\epsilon_j]\left(XH_\mu^{-1}\right)_i\left(H_\mu^{-1}X^T\right)_j$
- $\mathbb{E}[\epsilon_i\epsilon_j] = \sigma_\epsilon^2\delta_{ij}$
- $\sum_{i=1}^{N}\left(XH_\mu^{-1}\right)_i\left(H_\mu^{-1}X^T\right)_i = Tr\left[XH_\mu^{-2}X^T\right]$

Solution 1 - Generalization in A Teacher-Student Setup

$$
\begin{aligned}
\mathcal{R} &= \mathbb{E}\left\|\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)\mathbf{w}_{\text{true}}+\mathbf{H}_{\mu}^{-1}\mathbf{X}^{\top}\boldsymbol{\epsilon}\right\|^{2} \\
&= \mathbb{E}\left\|\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)\mathbf{w}_{\text{true}}\right\|^{2}+\mathbb{E}\|\mathbf{H}_{\mu}^{-1}\mathbf{X}^{\top}\boldsymbol{\epsilon}\|^{2} \\
&= \mathbb{E}\left[\mathbf{w}_{\text{true}}^{\top}\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)\mathbf{w}_{\text{true}}\right]+\mathbb{E}\left\|\boldsymbol{\epsilon}^{\top}\mathbf{X}\mathbf{H}_{\mu}^{-1}\mathbf{H}_{\mu}^{-1}\mathbf{X}^{\top}\boldsymbol{\epsilon}\right\| \\
&= \sum_{i,j=1}^{d}\mathbb{E}[w_{i,\text{true}}w_{j,\text{true}}]\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)_{i}^{\top}\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)_{j}+\sum_{i,j=1}^{N}\mathbb{E}[\epsilon_i\epsilon_j]\left(\mathbf{X}\mathbf{H}_{\mu}^{-1}\right)_{i}\left(\mathbf{H}_{\mu}^{-1}\mathbf{X}^{\top}\right)_{j} \\
&= \sum_{i,j=1}^{d}\sigma_w^2\delta_{ij}\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)_{i}^{\top}\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)_{j}+\sum_{i,j=1}^{N}\sigma_\epsilon^2\delta_{ij}\left(\mathbf{X}\mathbf{H}_{\mu}^{-1}\right)_{i}\left(\mathbf{H}_{\mu}^{-1}\mathbf{X}^{\top}\right)_{j} \\
&= \sigma_w^2\sum_{i=1}^{d}\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)_{i}^{\top}\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)_{i}+\sum_{i=1}^{N}\sigma_\epsilon^2\left(\mathbf{X}\mathbf{H}_{\mu}^{-1}\right)_{i}\left(\mathbf{H}_{\mu}^{-1}\mathbf{X}^{\top}\right)_{i} \\
&= \sigma_w^2\text{Tr}\left[\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)^{2}\right]+\sigma_\epsilon^2\text{Tr}\left[\mathbf{X}\mathbf{H}_{\mu}^{-2}\mathbf{X}^{\top}\right] \\
&= \sigma_w^2\text{Tr}\left[\left(\mathbf{H}_{\mu}^{-1}\mathbf{H}-\mathbf{I}\right)^{2}\right]+\sigma_\epsilon^2\text{Tr}\left[\mathbf{X}^{\top}\mathbf{X}\mathbf{H}_{\mu}^{-2}\right] \\
&= \sigma_w^2\text{Tr}\left[\left((\mathbf{H}+\mu\mathbf{I})^{-1}\mathbf{H}-\mathbf{I}\right)^{2}\right]+\sigma_\epsilon^2\text{Tr}\left[\mathbf{H}(\mathbf{H}+\mu\mathbf{I})^{-2}\right] \\
&= \sigma_w^2\text{Tr}\left[\left(\left(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\top}+\mu\mathbf{I}\right)^{-1}\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\top}-\mathbf{I}\right)^{2}\right]+\sigma_\epsilon^2\text{Tr}\left[\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\top}\left(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\top}+\mu\mathbf{I}\right)^{-2}\right] \\
&= \sigma_w^2\text{Tr}\left[\left(\mathbf{U}(\boldsymbol{\Lambda}+\mu\mathbf{I})^{-1}\mathbf{U}^{\top}\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\top}-\mathbf{I}\right)^{2}\right]+\sigma_\epsilon^2\text{Tr}\left[\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\top}\mathbf{U}(\boldsymbol{\Lambda}+\mu\mathbf{I})^{-2}\mathbf{U}^{\top}\right] \\
&= \sigma_w^2\text{Tr}\left[\left(\mathbf{U}(\boldsymbol{\Lambda}+\mu\mathbf{I})^{-1}\boldsymbol{\Lambda}\mathbf{U}^{\top}-\mathbf{I}\right)^{2}\right]+\sigma_\epsilon^2\text{Tr}\left[\mathbf{U}\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\mu\mathbf{I})^{-2}\mathbf{U}^{\top}\right] \\
&= \sigma_w^2\text{Tr}\left[\mathbf{U}\left((\boldsymbol{\Lambda}+\mu\mathbf{I})^{-1}\boldsymbol{\Lambda}-\mathbf{I}\right)^{2}\mathbf{U}^{\top}\right]+\sigma_\epsilon^2\text{Tr}\left[\mathbf{U}\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\mu\mathbf{I})^{-2}\mathbf{U}^{\top}\right] \\
&= \sigma_w^2\text{Tr}\left[\left((\boldsymbol{\Lambda}+\mu\mathbf{I})^{-1}\boldsymbol{\Lambda}-\mathbf{I}\right)^{2}\right]+\sigma_\epsilon^2\text{Tr}\left[\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\mu\mathbf{I})^{-2}\right] \\
&= \sigma_w^2\sum_{i=1}^{d}\left(\frac{\lambda_i}{\lambda_i+\mu}-1\right)^{2}+\sigma_\epsilon^2\sum_{i=1}^{d}\frac{\lambda_i}{(\lambda_i+\mu)^{2}} \\
&= \sum_{i=1}^{d}\frac{\sigma_w^2\mu^2+\sigma_\epsilon^2\lambda_i}{(\lambda_i+\mu)^{2}}
\end{aligned}
$$

Question 2 - "Typical" Generalization in Multilayer Neural Networks

---

We examine a "student" neural network $f_{\mathbf{w}}(\mathbf{x})$ with parameter vector $\mathbf{w}\in\mathbb{R}^{k}$ and input $\mathbf{x}\in\mathbb{R}^{d_0}$ used in a binary classification problem where the training set is $\mathcal{S}=\left\{\mathbf{x}^{(n)}\right\}_{n=1}^{N}$ sampled i.i.d. from P_X, where the binary (± 1) labels are generated by a "teacher" neural network $f_{\mathbf{w}_\star}(\mathbf{x})$ with the same architecture. To understand the "typical" generalization of the student, we examine the following "Guess and Check" algorithm to learn its weights: we randomly sample parameters vectors $\mathbf{w}_1,\mathbf{w}_2,\ldots$ i.i.d.from P_W, in which each parameter is sampled independently from a uniform distribution over $Q=\{-(q-1)/2,\ldots,-1,0,1,\ldots,(q-1)/2\}$ quantization levels, where $q=|Q|$ is an odd positive number. We do this until a stopping time t in which we perfectly fit the dataset: $\forall n: f_{\mathbf{w}_t}\left(\mathbf{x}^{(n)}\right)=f_{\mathbf{w}_\star}\left(\mathbf{x}^{(n)}\right)$. We examine a two-layer neural network with d_1 hidden neurons

$$
f_{\mathbf{w}}(\mathbf{x})=\text{sign}\left(\mathbf{w}_2^{\top}\left[\mathbf{W}_1\mathbf{x}\right]_{+}\right)
$$

where $[\cdot]_{+}$ is the ReLU activation function, the teacher has at most $d_1^{\star}<d_1$ non-zero neurons (i.e., the other $d_1-d_1^{\star}$ hidden neurons in the teacher to have all the incoming and outgoing weights equal to zero). Each of the teacher's weights are also in Q.

1. Calculate the probability $P_{\mathbf{w}\sim P_W}\left(\mathbf{w}=\mathbf{w}_\star\right)$.
2. Prove that

$$
(1)\ \ p_\star\triangleq P_{\mathbf{w}\sim P_W}\left(\forall\mathbf{x}: f_{\mathbf{w}}(\mathbf{x})=f_{\mathbf{w}_\star}(\mathbf{x})\right)\geq q^{-d_0d_1^{\star}-d_1}. \tag{1}
$$

3. Show that for any constant $T>0$, we can bound the probability of stopping time $t>T$ as

$$(2)\quad \lfloor T \rfloor \leq \frac{\log P\left(t>T\right)}{\log(1-p_{\star})}\,. \tag{2}$$

4. Prove the generalization bound:
   Theorem 1 *With probability* $(1-\eta)\,(1-\delta)$,

$$(3)\quad \epsilon < \frac{\left(d_0 d_1^{\star}+d_1\right)\log q+\log\frac{1}{\delta}+\log\log\frac{1}{\eta}}{N} \tag{3}$$

   Hint: Combine the results from previous sections, using the approximations $\lfloor T \rfloor \approx T$ and $\log(1-p_{\star}) \approx -p_{\star}$ (treat these approximations as exact), and the following basic generalization bound (which we learned in class):
   Theorem 2 *For any* $f \in |\mathcal{F}|$ $f \in |\mathcal{F}|$, *with probability* $1-\delta$,

$$(4)\quad \epsilon \triangleq \mathbb{P}_{\mathbf{x}}\left(f_{\mathbf{w}}\left(\mathbf{x}\right) \neq f_{\mathbf{w}_{\star}}\left(\mathbf{x}\right)\right) < \frac{\log|\mathcal{F}|+\log\frac{1}{\delta}}{N}\,. \tag{4}$$

5. Is the bound in eq. (3) better than the bound in eq. (4) in which $\mathcal{F}=\left\{f_{\mathbf{w}}:\mathbf{w}\in Q^{k}\right\}$ is the student hypothesis class (in which each parameter can have one of q values)? Explain and ignore the (negligble) $\log\log\frac{1}{\eta}$ term.

Solution 2 - "Typical" Generalization in Multilayer Neural Networks

---

1. To get a specific weights we set each of the k weights to recieve a specific value out of q possible values. Since the weights are i.i.d this entails

$$P_{\mathbf{w}\sim P_W}\left(\mathbf{w}=\mathbf{w}_{\star}\right)=q^{-k}\,.$$

2. In this case, we need to sample correctly all the input and output weights of the $d_1^{\star}$ hidden neurons in the teacher neural network. In addition we need to sample '0' in the output weights of all the $\left(d_1-d_1^{\star}\right)$ student neurons that are not in the teacher neurons. Therefore, we get

$$p_{\star} \geq q^{-d_0 d_1^{\star}-d_1}\,.$$

3. If $t>T$ it means we were not able to fit the training set for T times,

$$P\left(t>T\right)=\left(1-P_{\mathbf{w}\sim P_W}\left(\forall\mathbf{x}\in\mathcal{S}:f_{\mathbf{w}}\left(\mathbf{x}\right)=f_{\mathbf{w}_{\star}}\left(\mathbf{x}\right)\right)\right)^{T} \leq \left(1-p_{\star}\right)^{T},$$

   where in the inequality we used the fact the probability of achieving a perfect classification on the training set can be lower bounded by the probability of achieving the same function as the teacher. Taking the log of both sides of the inequality we get

$$\log\left(1-p_{\star}\right)^{T} \geq \log P\left(t>T\right) \Rightarrow T\log(1-p_{\star}) \geq \log P\left(t>T\right) \overset{\log(1-p_{\star})<0}{\Rightarrow} T \leq \frac{\log P\left(t>T\right)}{\log(1-p_{\star})}\,.$$

4. Denoting $\eta \triangleq P\left(t>T\right)$, we have that $1-\eta=P\left(t<T\right)$. So, with probability $1-\eta$, we have $f_{\mathbf{w}_t} \in \mathcal{F}_T=\{f_{\mathbf{w}_1},\ldots,f_{\mathbf{w}_T}\}$, and

$$\begin{aligned}\log|\mathcal{F}_T| = \log T &\leq \log\left(\frac{\log P\left(t>T\right)}{\log(1-p_{\star})}\right)=\log\left(\frac{\log\eta}{\log(1-p_{\star})}\right)\\ &\approx \log\left(\frac{\log\eta}{-p_{\star}}\right)=\log\log\frac{1}{\eta}-\log p_{\star} \leq \log\log\frac{1}{\eta}+\left(d_0 d_1^{\star}+d_1\right)\log q.\end{aligned}$$

   where in the first inequality we used eq. (2), and in the last equality we used eq. (1). Plugging this into the generalization bound (eq. (4)), we get the Theorem we aimed to prove.
5. With q quantization levels, and $k=d_0 d_1+d_1$ parameters we get in eq. (4)

$$\log|\mathcal{F}|=k\log q=\left(d_0 d_1+d_1\right)\log q.$$

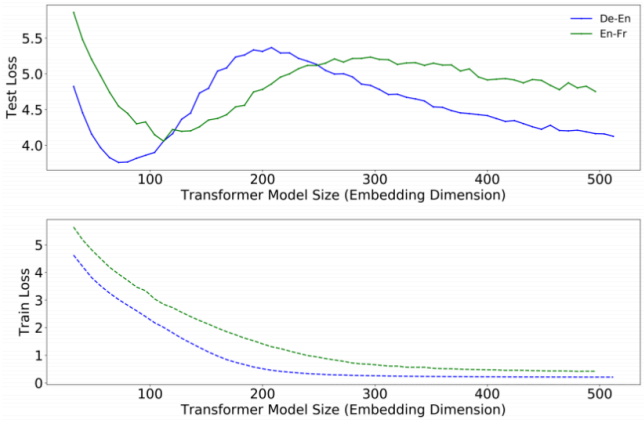
   This is larger than $\left(d_0 d_1^{\star}+d_1\right)\log q$ if $d_1^{\star}<d_1$ as we assume, and therefore leads to a worse generalization bound than in eq. (3).

Question 3 - Deep Double Descent

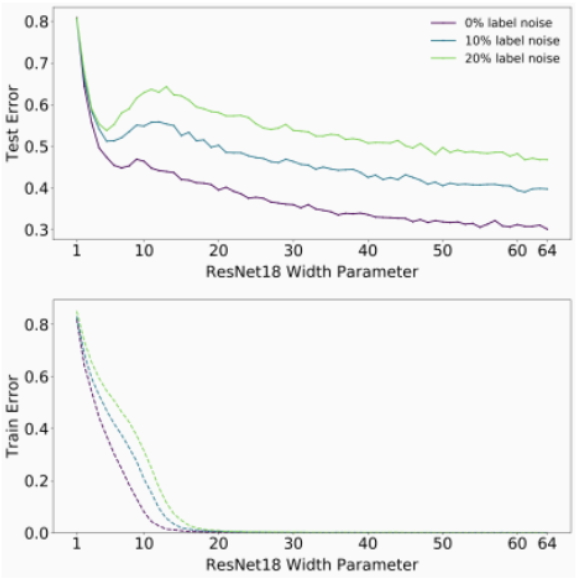
For the following plots:

1. Where is the critical point (the point of transition between the "Classical Regime" and "Modern Regime") of the deep double descent?
2. What type of double descent is shown? Explain. There can be more than one correct answer.

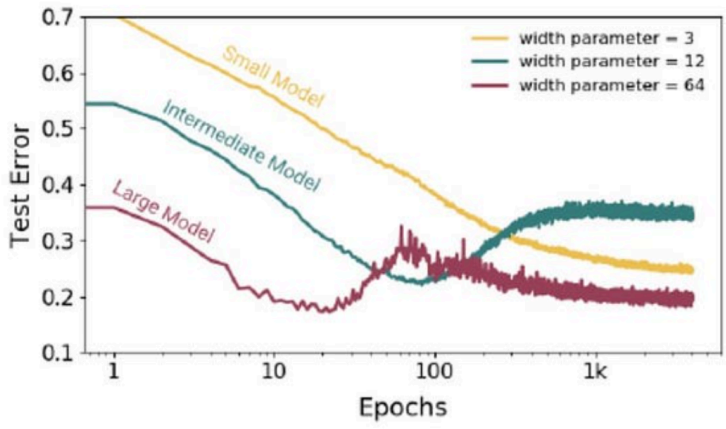
a.



b.



c.



Solution 3 - Deep Double Descent

a. 200-300

b. 10

Both Model-wise Double Descent

c. Epoch + model-wise double descent. Sufficiently large models can undergo a “double descent” behavior where test error first decreases then increases near the interpolation threshold, and then decreases again. In contrast, for “medium sized” models, for which training to completion will only barely reach a level of error close to 0, the test error as a function of training time will follow a classical U-like curve where it is better to stop early. Models that are too small to reach the approximation threshold will remain in the “under parameterized” regime where increasing train time monotonically decreases test error.

Question 4 - Initialization

Recall that in lecture 5 we were discussing how to calculate the initialization variance, and reached the conclusion that

$$\sigma_l = \frac{1}{\sqrt{\sum_j \mathbb{E}\left[\varphi^2(u_{l-1}[j])\right]}}$$

Show that for ReLU activation ($\varphi(z) = max(0, z)$), the optimal variance satisfies:

$$\sigma_l = \sqrt{\frac{2}{d_{l-1}}}$$

1. Under the assumption that the distribution of W is symmetric ($\rightarrow$ the distribution of u is symmetric).
2. Using the central limit theorem for large width.

Answer each section **separately** and assume the sections are independent.

All the notations are the same as in the lecture slides.

Solution 4 -Initialization

1. For ReLU:

$$\sigma_l = \frac{1}{\sqrt{\sum_j \mathbb{E}\left[\varphi^2(u_{l-1}[j])\right]}} = \frac{1}{\sqrt{d_{l-1}\mathbb{E}\left[\varphi^2(u_{l-1}[j])\right]}} = \frac{1}{\sqrt{d_{l-1}\cdot\frac{1}{2}}} = \sqrt{\frac{2}{d_{l-1}}},$$

   where the first transition is from symmetry and see below for the calculation of the expectancy.

2. For ReLU: $\varphi^2(z) = max(0, z^2)$, and $\mathbb{E}_{z\sim\mathcal{N}(0,1)}[max(0, z^2)] = \frac{1}{2}\mathbb{E}_{z\sim\mathcal{N}(0,1)}[z^2]$. We know that

$$Var(z) = \mathbb{E}[z^2] - (\mathbb{E}[z])^2$$

$$\rightarrow \mathbb{E}[z^2] = 1 + 0 = 1$$

$$\rightarrow \sigma_l = \frac{1}{\sqrt{d_{l-1}(\frac{1}{2}\cdot\mathbb{E}[z^2])}} = \sqrt{\frac{2}{d_{l-1}}}$$

Question 5 - Equivariance

Recall from lecture 6: A function $f : \mathbb{R}^d \to \mathbb{R}^d$ is equivariant if $f(\tau \cdot x) = \tau \cdot f(x)$ for all τ.

Let $f_w(x) = \phi(Wx)$ where ϕ is a component-wise non-linearity and $W \in \mathbb{R}^{d \times d}$. Prove that $f_w : \mathbb{R}^d \to \mathbb{R}^d$ is equivariant to transformation family H **if and only if**:

$$\forall \tau \in H, W[i,j] = W[\tau(i), \tau(j)]$$

- As in class, τ is an operator which re-arranges the terms in the vector it is operating on. $\tau(i) = j$ implies that component i is mapped to component j. In addition, $\tau \cdot x$ means we use τ on x.
- Assume one-by-one activations (Injective functions/one-by-one)

Solution 5 - Equivariance

Recall that ϕ operates **component-wise**.

1. $\rightarrow$: if $W[i,j] = W[\tau(i), \tau(j)]$ then:

$$f_w(\tau x) = \phi(W\tau x) = \phi\left(\begin{bmatrix} | \\ \sum_j W[i,j]x[\tau(j)] \\ | \end{bmatrix}\right) = \phi\left(\begin{bmatrix} | \\ \sum_j W[\tau(i),\tau(j)]x[\tau(j)] \\ | \end{bmatrix}\right)$$
$$= \tau\phi(Wx) = \tau f_w(x)$$

2. $\leftarrow$: if $f_w(\tau x) = \tau f_w(x)$ then:

$$\phi(W\tau x) = \phi\left(\begin{bmatrix} | \\ \sum_j W[i,j]x[\tau(j)] \\ | \end{bmatrix}\right) = \phi\left(\begin{bmatrix} | \\ \sum_j W[\tau(i),\tau(j)]x[\tau(j)] \\ | \end{bmatrix}\right)$$
$$\rightarrow W[i,j] = W[\tau(i), \tau(j)]$$

Question 6 -VGG Architecture

1. The VGG-11 CNN architecture consists of 11 convolution (CONV)/fully-connected (FC) layers (every CONV layer has the same padding and stride, every MAXPOOL layer is 2×2 and has padding of 0 and stride 2). Fill in the table. You need to **consider the bias**.

- CONVM-N: a convolutional layer of size $M \times M \times N$, where M is the kernel size and N is the number of filters. $stride = 1, padding = 1$.
- POOL2: 2×2 Max Pooling with $stride = 2$
  - In case the input of the layer is odd, you should round down. For example, if the output of the layer should be $3.5 \times 3.5 \times 3$, you should round to $3 \times 3 \times 3$ (i.e., ignore the last column of the input image when performing MaxPooling).
- FC-N: a fully connected layer with N neurons.

Layer	Output Dimension	Number of Parameters (Weights)
INPUT	224x224x3	0
CONV3-64	-	-
ReLU	-	-
POOL2	-	-
CONV3-128	-	-
ReLU	-	-
POOL2	-	-
CONV3-256	-	-

Layer	Output Dimension	Number of Parameters (Weights)
ReLU	-	-
CONV3-256	-	-
ReLU	-	-
POOL2	-	-
CONV3-512	-	-
ReLU	-	-
CONV3-512	-	-
ReLU	-	-
POOL2	-	-
CONV3-512	-	-
ReLU	-	-
CONV3-512	-	-
ReLU	-	-
POOL2	-	-
FC-4096	-	-
FC-4096	-	-
FC-1000	-	-
SOFTMAX	-	-

2. What is the total number of parameters? (use a calculator for this one)
3. What percentage of the weights are found in the fully-connected layers?

Solution 6 - VGG Architecture

1. Solution:

Layer	Output Dimension	Number of Parameters (Weights)
INPUT	224x224x3	0
CONV3-64	224×224×64	$(3 \times 3 \times 3 + 1)64 = 1,792$
ReLU	224×224×64	0
POOL2	112×112×64	0
CONV3-128	112×112×128	$(3 \times 3 \times 64 + 1)128 = 73,856$
ReLU	112×112×128	0
POOL2	56×56×128	0
CONV3-256	56×56×256	$(3 \times 3 \times 128 + 1)256 = 295,168$
ReLU	56×56×256	0
CONV3-256	56×56×256	$(3 \times 3 \times 256 + 1)256 = 590,080$
ReLU	56×56×256	0
POOL2	28×28×256	0
CONV3-512	28×28×512	$(3 \times 3 \times 256 + 1)512 = 1,180,160$

Layer	Output Dimension	Number of Parameters (Weights)
ReLU	28×28×512	0
CONV3-512	28×28×512	$(3 \times 3 \times 512 + 1)512 = 2,359,808$
ReLU	28×28×512	0
POOL2	14×14×512,	0
CONV3-512	14×14×512	$(3 \times 3 \times 512 + 1)512 = 2,359,808$
ReLU	14×14×512	0
CONV3-512	14×14×512	$(3 \times 3 \times 512 + 1)512 = 2,359,808$
ReLU	14×14×512	0
POOL2	7×7×512	0
FC-4096	4096	$(7 \times 7 \times 512 + 1)4096 = 102,764,544$
FC-4096	4096	$(4096 + 1) \times 4096 = 16,781,312$
FC-1000	1000	$(4096 + 1) \times 1000 = 4,097,000$
SOFTMAX	1000	0

2. 132,863,336
3. 93%

Credits

- Icons made by Becris from www.flaticon.com
- Icons from Icons8.com - https://icons8.com
- Datasets from Kaggle - https://www.kaggle.com/