Let's try to think about why PLA may work.

Let $n = n(t)$, according to the rule of PLA below, which formula is true?

$$\text{sign}\left(\mathbf{w}_t^T \mathbf{x}_n\right) \neq y_n, \quad \mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_n \mathbf{x}_n$$

1. $\mathbf{w}_{t+1}^T \mathbf{x}_n = y_n$
2. $\text{sign}(\mathbf{w}_{t+1}^T \mathbf{x}_n) = y_n$
3. $y_n \mathbf{w}_{t+1}^T \mathbf{x}_n \geq y_n \mathbf{w}_t^T \mathbf{x}_n$
4. $y_n \mathbf{w}_{t+1}^T \mathbf{x}_n < y_n \mathbf{w}_t^T \mathbf{x}_n$

Fun Time

Let's upper-bound T, the number of mistakes that PLA 'corrects'.

Define $R^2 = \max_n \|\mathbf{x}_n\|^2 \quad \rho = \min_n y_n \frac{\mathbf{w}_f^T}{\|\mathbf{w}_f\|} \mathbf{x}_n$

We want to show that $T \leq \square$. Express the upper bound $\square$ by the two terms above.

1. R/ρ
2. R^2/ρ^2
3. R/ρ^2
4. ρ^2/R^2

Reference Answer: 2

The maximum value of $\frac{\mathbf{w}_f^T}{\|\mathbf{w}_f\|} \frac{\mathbf{w}_t}{\|\mathbf{w}_t\|}$ is 1. Since T mistake corrections **increase the inner product by $\sqrt{T}$· constant**, the maximum number of corrected mistakes is $1/\text{constant}^2$.

Fun Time

Should we use pocket or PLA?

Since we do not know whether $\mathcal{D}$ is linear separable in advance, we may decide to just go with pocket instead of PLA. If $\mathcal{D}$ is actually linear separable, what's the difference between the two?

1. pocket on $\mathcal{D}$ is slower than PLA
2. pocket on $\mathcal{D}$ is faster than PLA
3. pocket on $\mathcal{D}$ returns a better g in approximating f than PLA
4. pocket on $\mathcal{D}$ returns a worse g in approximating f than PLA

Reference Answer: 1

Because pocket need to check whether $\mathbf{w}_{t+1}$ is better than $\hat{\mathbf{w}}$ in each iteration, it is slower than PLA. On linear separable $\mathcal{D}$, $\mathbf{w}_{\text{POCKET}}$ is the same as $\mathbf{w}_{\text{PLA}}$, both making no mistakes.

Fun Time

This is a popular 'brain-storming' problem, with a claim that 2% of the world's cleverest population can crack its 'hidden pattern'.

$$(5,3,2) \rightarrow 151022, \qquad (7,2,5) \rightarrow ?$$

It is like a 'learning problem' with $N = 1$, $\mathbf{x}_1 = (5,3,2)$, $y_1 = 151022$. Learn a hypothesis from the one example to predict on $\mathbf{x} = (7,2,5)$. What is your answer?

1. 151026
2. 143547
3. I need more examples to get the correct answer
4. there is no 'correct' answer

Reference Answer: (4)

Following the same nature of the no-free-lunch problems discussed, we cannot hope to be correct under this 'adversarial' setting. BTW, (2) is the designer's answer: the first two digits $= x_1 \cdot x_2$; the next two digits $= x_1 \cdot x_3$; the last two digits $= (x_1 \cdot x_2 + x_1 \cdot x_3 - x_2)$.

Fun Time

Let $\mu = 0.4$. Use Hoeffding's Inequality

$$\mathbb{P}\left[|\nu - \mu| > \epsilon\right] \leq 2\exp\left(-2\epsilon^2 N\right)$$

to bound the probability that a sample of 10 marbles will have $\nu \leq 0.1$. What bound do you get?

1. 0.67
2. 0.40
3. 0.33
4. 0.05

Reference Answer: (3)

Set $N = 10$ and $\epsilon = 0.3$ and you get the answer. BTW, (4) is the actual probability and Hoeffding gives only an upper bound to that.

Consider 4 hypotheses.

$$h_1(\mathbf{x}) = \text{sign}(x_1), \quad h_2(\mathbf{x}) = \text{sign}(x_2),$$
$$h_3(\mathbf{x}) = \text{sign}(-x_1), \quad h_4(\mathbf{x}) = \text{sign}(-x_2).$$

For any N and ϵ, which of the following statement is not true?

1. the **BAD** data of h_1 and the **BAD** data of h_2 are exactly the same
2. the **BAD** data of h_1 and the **BAD** data of h_3 are exactly the same
3. $\mathbb{P}_{\mathcal{D}}$[**BAD** for some h_k] $\leq 8\exp\left(-2\epsilon^2 N\right)$
4. $\mathbb{P}_{\mathcal{D}}$[**BAD** for some h_k] $\leq 4\exp\left(-2\epsilon^2 N\right)$

Reference Answer: (1)

The important thing is to note that (2) is true, which implies that (4) is true if you revisit the union bound. Similar ideas will be used to conquer the $M = \infty$ case.

Data size: how large do we need?

One way to use the inequality

$$\mathbb{P}\left[\left|E_{\text{in}}(g) - E_{\text{out}}(g)\right| > \epsilon\right] \leq \underbrace{2 \cdot M \cdot \exp\left(-2\epsilon^2 N\right)}_{\delta}$$

is to pick a tolerable difference ϵ as well as a tolerable **BAD** probability δ, and then gather data with size (N) large enough to achieve those tolerance criteria. Let $\epsilon = 0.1$, $\delta = 0.05$, and $M = 100$. What is the data size needed?

1. 215
2. 415
3. 615
4. 815

Reference Answer: (2)

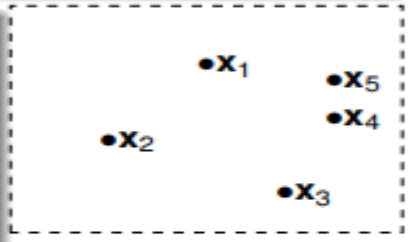
We can simply express N as a function of those 'known' variables. Then, the needed $N = \frac{1}{2\epsilon^2} \ln \frac{2M}{\delta}$.

What is the effective number of lines for five inputs $\in \mathbb{R}^2$?

1. 14
2. 16
3. 22
4. 32

Reference Answer: (3)

If you put the inputs roughly around a circle, you can then pick any consecutive inputs to be on one side of the line, and the other inputs to be on the other side. The procedure leads to effectively 22 kinds of lines, which is **much smaller than** $2^5 = 32$. You shall find it difficult to generate more kinds by varying the inputs, and we will give a formal proof in future lectures.



Fun Time

Consider positive **and negative** rays as $\mathcal{H}$, which is equivalent to the perceptron hypothesis set in 1D. The hypothesis set is often called **'decision stump'** to describe the shape of its hypotheses. What is the growth function $m_{\mathcal{H}}(N)$?

1. N
2. $N+1$
3. $2N$
4. 2^N

Reference Answer: 3

Two dichotomies when threshold in each of the $N-1$ 'internal' spots; two dichotomies for the all-$\circ$ and all-$\times$ cases.

3

Fun Time

Consider positive **and negative** rays as $\mathcal{H}$, which is equivalent to the perceptron hypothesis set in 1D. As discussed in an earlier quiz question, the growth function $m_{\mathcal{H}}(N) = 2N$. What is the minimum break point for $\mathcal{H}$?

1. 1
2. 2
3. 3
4. 4

Reference Answer: 3

At $k = 3$, $m_{\mathcal{H}}(k) = 6$ while $2^k = 8$.

+

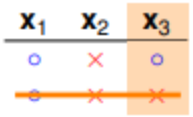
Fun Time

When minimum break point $k = 1$, what is the maximum possible $m_{\mathcal{H}}(N)$ when $N = 3$?

1. 1
2. 2
3. 4
4. 8

Reference Answer: (1)

Because $k = 1$, the hypothesis set cannot even shatter one point. Thus, every 'column' of the table cannot contain both $\circ$ and $\times$. Then, after excluding the first dichotomy, it is not possible to include any other different dichotomy. Thus, the maximum possible $m_{\mathcal{H}}(N)$ is 1.



Fun Time

For the 2D perceptrons, which of the following claim is true?

1. minimum break point $k = 2$
2. $m_{\mathcal{H}}(4) = 15$
3. $m_{\mathcal{H}}(N) < B(N, k)$ when $N = k =$ minimum break point
4. $m_{\mathcal{H}}(N) > B(N, k)$ when $N = k =$ minimum break point

Reference Answer: (3)

As discussed previously, minimum break point for 2D perceptrons is 4, with $m_{\mathcal{H}}(4) = 14$. Also, note that $B(4, 4) = 15$. So bounding function $B(N, k)$ can be 'loose' in bounding $m_{\mathcal{H}}(N)$.

Fun Time

For 1D perceptrons (positive and negative rays), we know that $m_{\mathcal{H}}(N) = 2N$. Let k be the minimum break point. Which of the following is not true?

1. $k = 3$
2. for some integers $N > 0$, $m_{\mathcal{H}}(N) = \sum_{i=0}^{k-1} \binom{N}{i}$
3. for all integers $N > 0$, $m_{\mathcal{H}}(N) = \sum_{i=0}^{k-1} \binom{N}{i}$
4. for all integers $N > 2$, $m_{\mathcal{H}}(N) < \sum_{i=0}^{k-1} \binom{N}{i}$

Reference Answer: (3)

The proof is generally trivial by listing the definitions. For (2), $N = 1$ or 2 gives the equality. One thing to notice is (4): the upper bound can be 'loose'.

Fun Time

For positive rays, $m_{\mathcal{H}}(N) = N + 1$. Plug it into the VC bound for $\epsilon = 0.1$ and $N = 10000$. What is VC bound of BAD events?

$$\mathbb{P}\left[\exists h \in \mathcal{H} \text{ s.t. } \left|E_{\text{in}}(h) - E_{\text{out}}(h)\right| > \epsilon\right] \leq 4 m_{\mathcal{H}}(2N) \exp\left(-\frac{1}{8}\epsilon^2 N\right)$$

1. 2.77×10^{-87}
2. 5.54×10^{-83}
3. 2.98×10^{-1}
4. 2.29×10^{2}

Reference Answer: (3)

Simple calculation. Note that the BAD probability bound is not very small even with 10000 examples.

If there is a set of N inputs that cannot be shattered by $\mathcal{H}$. Based only on this information, what can we conclude about $d_{\text{vc}}(\mathcal{H})$?

1. $d_{\text{vc}}(\mathcal{H}) > N$
2. $d_{\text{vc}}(\mathcal{H}) = N$
3. $d_{\text{vc}}(\mathcal{H}) < N$
4. no conclusion can be made

Reference Answer: (4)

It is possible that there is another set of N inputs that can be shattered, which means $d_{\text{vc}} \geq N$. It is also possible that no set of N input can be shattered, which means $d_{\text{vc}} < N$. Neither cases can be ruled out by one non-shattering set.

Extra Fun Time

What statement below shows that $d_{\text{vc}} \geq d + 1$?

1. There are some $d + 1$ inputs we can shatter.
2. We can shatter any set of $d + 1$ inputs.
3. There are some $d + 2$ inputs we cannot shatter.
4. We cannot shatter any set of $d + 2$ inputs.

Reference Answer: (1)

d_{vc} is the maximum that $m_{\mathcal{H}}(N) = 2^N$, and $m_{\mathcal{H}}(N)$ is the most number of dichotomies of N inputs. So if we can find 2^{d+1} dichotomies on *some* $d + 1$ inputs, $m_{\mathcal{H}}(d + 1) = 2^{d+1}$ and hence $d_{\text{vc}} \geq d + 1$.

Extra Fun Time

What statement below shows that $d_{vc} \leq d + 1$?

1. There are some $d + 1$ inputs we can shatter.
2. We can shatter any set of $d + 1$ inputs.
3. There are some $d + 2$ inputs we cannot shatter.
4. We cannot shatter any set of $d + 2$ inputs.

Reference Answer: 4

d_{vc} is the maximum that $m_{\mathcal{H}}(N) = 2^N$, and $m_{\mathcal{H}}(N)$ is the most number of dichotomies of N inputs. So if we cannot find 2^{d+2} dichotomies on *any* $d + 2$ inputs (i.e. break point), $m_{\mathcal{H}}(d + 2) < 2^{d+2}$ and hence $d_{vc} < d + 2$. That is, $d_{vc} \leq d + 1$.

Fun Time

Based on the proof above, what is d_{vc} of 1126-D perceptrons?

1. 1024
2. 1126
3. 1127
4. 6211

Reference Answer: 3

Well, too much fun for this section! :-)

Fun Time

Origin-crossing Hyperplanes are essentially perceptrons with w_0 fixed at 0. Make a guess about the d_{vc} of origin-crossing hyperplanes in $\mathbb{R}^d$.

1. 1
2. d
3. $d+1$
4. ∞

Reference Answer: 2

The proof is almost the same as proving the d_{vc} for usual perceptrons, but it is the **intuition** ($d_{vc} \approx$ #free parameters) that you shall use to answer this quiz.

Consider the VC Bound below. How can we decrease the probability of getting **BAD** data?

$$\mathbb{P}_{\mathcal{D}}\left[\left|E_{in}(g) - E_{out}(g)\right| > \epsilon\right] \quad \leq \quad 4(2N)^{d_{vc}} \exp\left(-\frac{1}{8}\epsilon^2 N\right)$$

1. decrease model complexity d_{vc}
2. increase data size N a lot
3. increase generalization error tolerance ϵ
4. all of the above

Reference Answer: 4

Congratulations on being Master of VC bound! :-)

Fun Time

Let's revisit PLA/pocket. Which of the following claim is true?

1. In practice, we should try to compute if $\mathcal{D}$ is linear separable before deciding to use PLA.
2. If we know that $\mathcal{D}$ is not linear separable, then the target function f must not be a linear function.
3. If we know that $\mathcal{D}$ is linear separable, then the target function f must be a linear function.
4. None of the above

Reference Answer: (4)

(1) After computing if $\mathcal{D}$ is linear separable, we shall know $\mathbf{w}^*$ and then there is no need to use PLA. (2) What about noise? (3) What about 'sampling luck'? :-)

Consider the following $P(y|\mathbf{x})$ and $\text{err}(\tilde{y}, y) = |\tilde{y} - y|$. Which of the following is the ideal mini-target $f(\mathbf{x})$?

$$P(y = 1|\mathbf{x}) = 0.10, P(y = 2|\mathbf{x}) = 0.35,$$
$$P(y = 3|\mathbf{x}) = 0.15, P(y = 4|\mathbf{x}) = 0.40.$$

1. $2.5 =$ average within $\mathcal{Y} = \{1, 2, 3, 4\}$
2. $2.85 =$ weighted mean from $P(y|\mathbf{x})$
3. $3 =$ weighted median from $P(y|\mathbf{x})$
4. $4 = \text{argmax}\, P(y|\mathbf{x})$

Reference Answer: (3)

For the 'absolute error', the weighted median provably results in the minimum average err.

Consider err below for CIA. What is $E_{\text{in}}(g)$ when using this err?

		g: +1	g: -1
f	+1	0	1
	-1	1000	0

1. $\frac{1}{N}\sum_{n=1}^{N}[\![y_n \neq g(\mathbf{x}_n)]\!]$
2. $\frac{1}{N}\left(\sum_{y_n=+1}[\![y_n \neq g(\mathbf{x}_n)]\!] + 1000\sum_{y_n=-1}[\![y_n \neq g(\mathbf{x}_n)]\!]\right)$
3. $\frac{1}{N}\left(\sum_{y_n=+1}[\![y_n \neq g(\mathbf{x}_n)]\!] - 1000\sum_{y_n=-1}[\![y_n \neq g(\mathbf{x}_n)]\!]\right)$
4. $\frac{1}{N}\left(1000\sum_{y_n=+1}[\![y_n \neq g(\mathbf{x}_n)]\!] + \sum_{y_n=-1}[\![y_n \neq g(\mathbf{x}_n)]\!]\right)$

Reference Answer: 2

When $y_n = -1$, the false positive made on such $(\mathbf{x}_n, y_n)$ is penalized 1000 times more!

Consider the CIA cost matrix. If there are 10 examples with $y_n = -1$ (intruder) and 999,990 examples with $y_n = +1$ (you). What would $E_{\text{in}}^{\text{w}}(h)$ be for a constant $h(\mathbf{x})$ that always returns $+1$?

		$h(\mathbf{x})$: +1	$h(\mathbf{x})$: -1
y	+1	0	1
	-1	1000	0

1. 0.001
2. 0.01
3. 0.1
4. 1

Reference Answer: 2

While the quiz is a simple evaluation, it is not uncommon that the data is very **unbalanced** for such an application. Properly 'setting' the weights can be used to avoid the lazy constant prediction.

Fun Time

Based on our discussion, for data of fixed size, which of the following situation is relatively of the lowest risk of overfitting?

1. small noise, fitting from small d_{vc} to median d_{vc}
2. small noise, fitting from small d_{vc} to large d_{vc}
3. large noise, fitting from small d_{vc} to median d_{vc}
4. large noise, fitting from small d_{vc} to large d_{vc}

Reference Answer: (1)

Two causes of overfitting are noise and excessive d_{vc}. So if both are relatively 'under control', the risk of overfitting is smaller.

Fun Time

When having limited data, in which of the following case would learner *R* perform better than learner *O*?

1. limited data from a 10-th order target function with some noise
2. limited data from a 1126-th order target function with no noise
3. limited data from a 1126-th order target function with some noise
4. all of the above

Reference Answer: (4)

We discussed about (1) and (2), but you shall be able to 'generalize' :-) that *R* also wins in the more difficult case of (3).

Fun Time

Consider the target function being $\sin(1126x)$ for $x \in [0, 2\pi]$. When x is uniformly sampled from the range, and we use all possible linear hypotheses $h(x) = w \cdot x$ to approximate the target function with respect to the squared error, what is the level of deterministic noise for each x?

1. $|\sin(1126x)|$
2. $|\sin(1126x) - x|$
3. $|\sin(1126x) + x|$
4. $|\sin(1126x) - 1126x|$

Reference Answer: 1

You can try a few different w and convince yourself that the best hypothesis h^* is $h^*(x) = 0$. The deterministic noise is the difference between f and h^*.

Fun Time

Assume we know that $f(x)$ is symmetric for some 1D regression application. That is, $f(x) = f(-x)$. One possibility of using the knowledge is to consider symmetric hypotheses only. On the other hand, you can also generate virtual examples from the original data $\{(x_n, y_n)\}$ as hints. What virtual examples suit your needs best?

1. $\{(x_n, -y_n)\}$
2. $\{(-x_n, -y_n)\}$
3. $\{(-x_n, y_n)\}$
4. $\{(2x_n, 2y_n)\}$

Reference Answer: 3

We want the virtual examples to encode the invariance when $x \to -x$.

For $Q \geq 1$, which of the following hypothesis (weight vector $\mathbf{w} \in \mathbb{R}^{Q+1}$) is not in the regularized hypothesis set $\mathcal{H}(1)$?

1. $\mathbf{w}^T = [0, 0, \ldots, 0]$
2. $\mathbf{w}^T = [1, 0, \ldots, 0]$
3. $\mathbf{w}^T = [1, 1, \ldots, 1]$
4. $\mathbf{w}^T = \left[\sqrt{\frac{1}{Q+1}}, \sqrt{\frac{1}{Q+1}}, \ldots, \sqrt{\frac{1}{Q+1}}\right]$

Reference Answer: (3)

The squared length of $\mathbf{w}$ in (3) is $Q + 1$, which is not ≤ 1.

Fun Time

When would $\mathbf{w}_{\text{REG}}$ equal $\mathbf{w}_{\text{LIN}}$?

1. $\lambda = 0$
2. $C = \infty$
3. $C \geq \|\mathbf{w}_{\text{LIN}}\|^2$
4. all of the above

Reference Answer: (4)

(1) and (2) shall be easy; (3) means that there are effectively no constraint on $\mathbf{w}$, hence the equivalence.

Fun Time

Consider the weight-decay regularization with regression. When increasing λ in $\mathcal{A}$, what would happen with $d_{\text{EFF}}(\mathcal{H}, \mathcal{A})$?

1. $d_{\text{EFF}} \uparrow$
2. $d_{\text{EFF}} \downarrow$
3. $d_{\text{EFF}} = d_{\text{VC}}(\mathcal{H})$ and does not depend on λ
4. $d_{\text{EFF}} = 1126$ and does not depend on λ

Reference Answer: (2)

larger λ
$\Longleftrightarrow$ smaller C
$\Longleftrightarrow$ smaller $\mathcal{H}(C)$
$\Longleftrightarrow$ smaller d_{EFF}

Fun Time

Consider using a regularizer $\Omega(\mathbf{w}) = \sum_{q=0}^{Q} 2^q w_q^2$ to work with Legendre polynomial regression. Which kind of hypothesis does the regularizer prefer?

1. symmetric polynomials satisfying $h(x) = h(-x)$
2. low-dimensional polynomials
3. high-dimensional polynomials
4. no specific preference

Reference Answer: 2

There is a higher 'penalty' for higher-order terms, and hence the regularizer prefers low-dimensional polynomials.

For $\mathcal{X} = \mathbb{R}^d$, consider two hypothesis sets, $\mathcal{H}_+$ and $\mathcal{H}_-$. The first hypothesis set contains all perceptrons with $w_1 \geq 0$, and the second hypothesis set contains all perceptrons with $w_1 \leq 0$. Denote g_+ and g_- as the minimum-E_{in} hypothesis in each hypothesis set, respectively. Which statement below is true?

1. If $E_{\text{in}}(g_+) < E_{\text{in}}(g_-)$, then g_+ is the minimum-E_{in} hypothesis of all perceptrons in $\mathbb{R}^d$.
2. If $E_{\text{test}}(g_+) < E_{\text{test}}(g_-)$, then g_+ is the minimum-E_{test} hypothesis of all perceptrons in $\mathbb{R}^d$.
3. The two hypothesis sets are disjoint.
4. None of the above

Reference Answer: 1

Note that the two hypothesis sets are not disjoint (sharing '$w_1 = 0$' perceptrons) but their union is all perceptrons.

For a learning model that takes N^2 seconds of training when using N examples, what is the total amount of seconds needed when running the whole validation procedure with $K = \frac{N}{5}$ on 25 such models with different parameters to get the final g_{m^*}?

1. $6N^2$
2. $17N^2$
3. $25N^2$
4. $26N^2$

Reference Answer: (2)

To get all the g_m^-, we need $\frac{16}{25}N^2 \cdot 25$ seconds.
Then to get g_{m^*}, we need another N^2 seconds.
So in total we need $17N^2$ seconds.

Fun Time

Consider three examples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3)$ with $y_1 = 1$, $y_2 = 5$, $y_3 = 7$. If we use E_{loocv} to estimate the performance of a learning algorithm that predicts with the average y value of the data set—the optimal constant prediction with respect to the squared error. What is E_{loocv} (squared error) of the algorithm?

1. 0
2. $\frac{56}{9}$
3. $\frac{60}{9}$
4. 14

Reference Answer: (4)

This is based on a simple calculation of $e_1 = (1-6)^2, e_2 = (5-4)^2, e_3 = (7-3)^2$.

For a learning model that takes N^2 seconds of training when using N examples, what is the total amount of seconds needed when running 10-fold cross validation on 25 such models with different parameters to get the final g_{m^*}?

1. $\frac{47}{2}N^2$
2. $47N^2$
3. $\frac{407}{2}N^2$
4. $407N^2$

Reference Answer: 3

To get all the E_{cv}, we need $\frac{81}{100}N^2 \cdot 10 \cdot 25$ seconds. Then to get g_{m^*}, we need another N^2 seconds. So in total we need $\frac{407}{2}N^2$ seconds.

Fun Time

For a deep NNet for written character recognition from raw pixels, which type of features are more likely extracted after the first hidden layer?

1. pixels
2. strokes
3. parts
4. digits

Reference Answer: 2

Simple strokes are likely the 'next-level' features that can be extracted from raw pixels.

Fun Time

Suppose training a d-$\tilde{d}$-d autoencoder with backprop takes approximately $c \cdot d \cdot \tilde{d}$ seconds. Then, what is the total number of seconds needed for pre-training a d-$d^{(1)}$-$d^{(2)}$-$d^{(3)}$-1 deep NNet?

1. $c\left(d + d^{(1)} + d^{(2)} + d^{(3)} + 1\right)$
2. $c\left(d \cdot d^{(1)} \cdot d^{(2)} \cdot d^{(3)} \cdot 1\right)$
3. $c\left(dd^{(1)} + d^{(1)}d^{(2)} + d^{(2)}d^{(3)} + d^{(3)}\right)$
4. $c\left(dd^{(1)} \cdot d^{(1)}d^{(2)} \cdot d^{(2)}d^{(3)} \cdot d^{(3)}\right)$

Reference Answer: 3

Each $c \cdot d^{(\ell-1)} \cdot d^{(\ell)}$ represents the time for pre-training with one autoencoder to determine one layer of the weights.

Fun Time

Which of the following cannot be viewed as a regularization technique?

1. hint the model with artificially-generated noisy data
2. stop gradient descent early
3. add a weight elimination regularizer
4. all the above are regularization techniques

Reference Answer: (4)

(1) is our new friend for regularization, while (2) and (3) are old friends.

Fun Time

When solving the optimization problem

$$\max_{\mathbf{v}} \sum_{n=1}^{N} \mathbf{v}^T \mathbf{x}_n \mathbf{x}_n^T \mathbf{v} \text{ subject to } \mathbf{v}^T \mathbf{v} = 1,$$

we know that the optimal $\mathbf{v}$ is the 'topmost' eigenvector that corresponds to the 'topmost' eigenvalue λ of $\mathrm{X}^T\mathrm{X}$. Then, what is the optimal objective value of the optimization problem?

1. λ^1
2. λ^2
3. λ^3
4. λ^4

Reference Answer: (1)

The objective value of the optimization problem is simply $\mathbf{v}^T\mathrm{X}^T\mathrm{X}\mathbf{v}$, which is $\lambda\mathbf{v}^T\mathbf{v}$ and you know what $\mathbf{v}^T\mathbf{v}$ must be.

Fun Time

Let $g_0(\mathbf{x}) = +1$. Which of the following $(\alpha_0, \alpha_1, \alpha_2)$ allows $G(\mathbf{x}) = \text{sign}\left(\sum_{t=0}^{2} \alpha_t g_t(\mathbf{x})\right)$ to implement $\text{OR}(g_1, g_2)$?

1. $(-3, +1, +1)$
2. $(-1, +1, +1)$
3. $(+1, +1, +1)$
4. $(+3, +1, +1)$

Reference Answer: 3

You can easily verify with all four possibilities of $(g_1(\mathbf{x}), g_2(\mathbf{x}))$.

Fun Time

How many weights $\{w_{ij}^{(\ell)}\}$ are there in a 3-5-1 NNet?

1. 9
2. 15
3. 20
4. 26

Reference Answer: 4

There are $(3+1) \times 5$ weights in $w_{ij}^{(1)}$, and $(5+1) \times 1$ weights in $w_{jk}^{(2)}$.

Fun Time

According to $\frac{\partial e_n}{\partial w_{i1}^{(L)}} = -2\left(y_n - s_1^{(L)}\right) \cdot \left(x_i^{(L-1)}\right)$ when would $\frac{\partial e_n}{\partial w_{i1}^{(L)}} = 0$?

1. $y_n = s_1^{(L)}$
2. $x_i^{(L-1)} = 0$
3. $s_i^{(L-1)} = 0$
4. all of the above

Reference Answer: 4

Note that $x_i^{(L-1)} = \tanh(s_i^{(L-1)}) = 0$ if and only if $s_i^{(L-1)} = 0$.

Fun Time

For the weight elimination regularizer $\sum \frac{\left(w_{ij}^{(\ell)}\right)^2}{1+\left(w_{ij}^{(\ell)}\right)^2}$, what is $\frac{\partial \text{regularizer}}{\partial w_{ij}^{(\ell)}}$?

1. $2w_{ij}^{(\ell)} / \left(1 + \left(w_{ij}^{(\ell)}\right)^2\right)^1$
2. $2w_{ij}^{(\ell)} / \left(1 + \left(w_{ij}^{(\ell)}\right)^2\right)^2$
3. $2w_{ij}^{(\ell)} / \left(1 + \left(w_{ij}^{(\ell)}\right)^2\right)^3$
4. $2w_{ij}^{(\ell)} / \left(1 + \left(w_{ij}^{(\ell)}\right)^2\right)^4$

Reference Answer: 2

Too much calculus in this class, huh? :-)