Introduction to Machine Learning (67577)

Exercise 6 - PCA, kernels, SGD and DL

Dor Roter
208772251

June 28, 2021

1 PCA

1. Let $v \in \mathbb{R}^d$,then $\langle v, X \rangle$ is the projection of X to the plan defined by v, now let us show that the vector used for embedding X into single dimension, maximizes the covariance.

$$\begin{aligned} Var\left(\langle v, X \rangle\right) =& Var\left(v^T X\right) = \mathbb{E}_X\left[\left(v^T X - \mathbb{E}_X\left[v^T X\right]\right)^2\right] = \\ & \mathbb{E}_X\left[\left(v^T\left(X - \mathbb{E}_X\left[X\right]\right)\right)\left(v^T\left(X - \mathbb{E}_X\left[X\right]\right)^T\right)\right] = \\ & \mathbb{E}_X\left[v^T\left(X - \mathbb{E}_X\left[X\right]\right)\left(X - \mathbb{E}_X\left[X\right]\right)^T v\right] = \\ & v^T \mathbb{E}_X\left[\left(X - \mathbb{E}_X\left[X\right]\right)\left(X - \mathbb{E}_X\left[X\right]\right)^T\right] v = \\ & = v^T Var\left(X\right) v = v^T \Sigma v \end{aligned}$$

Therefore maximizing the projected variance in respect to v can be achieved by using the lagranginan $g\left(v\right) = 1 - v^T v$:

$$\hat{v} = \underset{v \in \mathbb{R}^d, ||v||=1}{argmax}\ v^T \Sigma v = \underset{v \in \mathbb{R}^d}{argmax} \mathcal{L}(v, \lambda) = \underset{v \in \mathbb{R}^d}{argmax} v^T \Sigma v + \lambda g\left(v\right)$$

Now as both $g\left(\cdot\right)$ and $v^T \Sigma v$ are clearly concave functions, so their linear combination $\mathcal{L}(v, \lambda)$ must also be concave, and so in order to find a maximum we can follow Fermat's theorem:

$$\begin{aligned} \frac{\partial}{\partial v}\mathcal{L}(v, \lambda) &= 2\Sigma v - 2\lambda v = 0 \\ \frac{\partial}{\partial \lambda}\mathcal{L}(v, \lambda) &= 1 - v^T v = 0 \end{aligned}$$

Thus both of the following terms must hold for a maximizer $\hat{v}$:

$$\begin{aligned} &\Sigma v = \lambda v \\ &v^T v = 1 \Leftrightarrow ||v|| = 1 \end{aligned}$$

And so

$$\Sigma v = \lambda v \Leftrightarrow v^T \Sigma v = v^T \lambda v = \lambda v^T v = \lambda$$

$$\Downarrow$$

$$\max_{||v||=1} Var\left(\langle v, X \rangle\right) = \max_{||v||=1} v^T \Sigma v = \max_{\lambda \text{is an eigenvalue of X}} \lambda$$

Therefore the $\hat{v}$ for which X's projection's variance is maximal is an eigenvector of the maximal eigenvalue λ_1, exactly the PCA's one dimension embeding - and so no vector $v \in \mathbb{R}^d$ might have a large variance than the PCA embedding of X into a single dimension. $\square$

2 Kernels

2. Let $k(x, x')$ be a given valid kernel, let us define a new kernel $\tilde{k}$ by:

$$\tilde{k}(x, x') = \frac{k(x, x')}{\sqrt{k(x, x) \cdot k(x', x')}}$$

Firstly, it is easy to note that it is in-fact normalized, as:

$$\tilde{k}(x, x) = \frac{k(x, x)}{\sqrt{k(x, x) \cdot k(x, x)}} = \frac{k(x, x)}{k(x, x)} = 1$$

Next, as we have seen in recitation, for any function f if k is a valid kernel then $k'(x, x') = f(x)k(x, x')f(x')$ is also a valid kernel and so defining $f(x) = \frac{1}{\sqrt{k(x,x)}}$:

$$\tilde{k}(x, x') = \frac{k(x, x')}{\sqrt{k(x, x) \cdot k(x', x')}} = \frac{k(x, x')}{\sqrt{k(x, x)} \cdot \sqrt{k(x', x')}} = \frac{1}{\sqrt{k(x, x)}} k(x, x') \frac{1}{\sqrt{k(x', x')}} = f(x)k(x, x')f(x')$$

Thus $\tilde{k}$ is a valid, normalized kernel. □

3. Let $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{Y} = \left\{\left(\pm 1,\right) 1\right\}$and $S = \left\{\left(\begin{pmatrix}-2\\0\end{pmatrix}, 1\right), \left(\begin{pmatrix}0\\0\end{pmatrix}, -1\right), \left(\begin{pmatrix}2\\0\end{pmatrix}, 1\right)\right\}$ thus a linear seprator hypothesis class over $\mathbb{R}^2$ is defined by some linear 2d line, and so it is impossible to provide a linear seprator in $\mathbb{R}^2$ for the provided samples, as any line crossing would group together at best x_1, x_2 or x_2, x_3.
Let $\psi : \mathbb{R}^2 \to \mathbb{R}^3$ be defined by $\psi(x) = \begin{pmatrix} x_1 \\ x_2 \\ (x_1 + x_2)^2 \end{pmatrix}$, and so now $S_\psi = \left\{\left(\begin{pmatrix}-2\\0\\4\end{pmatrix}, 1\right), \left(\begin{pmatrix}0\\0\\0\end{pmatrix}, -1\right), \left(\begin{pmatrix}2\\0\\4\end{pmatrix}, -1\right)\right\}$
and we can linearly seprate the samples using the plane defined by $w = \begin{pmatrix}0\\0\\1/2\end{pmatrix}$ and $b = -1$. □

3 Convex optimization

4. (a) Let $f_i : V \to \mathbb{R}$for $i \in [m]$ be a convex function and γ_i be a non-negative scalar.
Let us show that $g(u) = \sum_{i=1}^{m} \gamma_i f_i(u)$ is convex.
Let $u, v \in V$:

$$\begin{aligned} g(u) + \nabla g(u)^T (v-u) &= \sum_{i=1}^{m} \gamma_i f_i(u) + \left(\sum_{i=1}^{m} \nabla \gamma_i f_i(u) \right)^T (v-u) \\ &= \sum_{i=1}^{m} \gamma_i f_i(u) + \sum_{i=1}^{m} \nabla \gamma_i f_i(u)^T (v-u) = \sum_{i=1}^{m} \left(\gamma_i f_i(u) + \nabla \gamma_i f_i(u)^T (v-u) \right) \\ &= \sum_{i=1}^{m} \gamma_i \left(f_i(u) + \nabla f_i(u)^T (v-u) \right) \underset{f_i \text{ convexity}}{\leq} \sum_{i=1}^{m} \gamma_i f(v) = g(v) \end{aligned}$$

And thus by the fist order condition g is convex. □

(b) Let $f(x) = (x-4)^2$ and $g(x) = x^2$, then clearly both f and g are convex as their second order derivatives is 2, and thus clearly non-negative.
Yet $h(x) = f(g(x)) = \left(x^2 - 4\right)^2$ is non-convex, as:

$$\begin{aligned} h'(x) &= 2\left(x^2-4\right)2x = 4x^3 - 16x \\ h''(x) &= 12x^2 - 16 \end{aligned}$$

And so for any $x \in \left(-\sqrt{4/3}, \sqrt{4/3}\right)$ the second order derivative of h is negative and thus it is not convex. □

(c) "⇒": Let $f : C \to \mathbb{R}$ be a convex function defined over a convex set C.
Then for any $(u, t_1), (v, t_2) \in epi(f)$, and $\alpha \in [0,1]$ it holds that:

$$f(\alpha u + (1-\alpha) v) \underset{f \text{ is convex}}{\leq} \alpha f(u) + (1-\alpha) f(v) \underset{(u,t_1),(v,t_2) \in epi(f)}{\leq} \alpha t_1 + (1-\alpha) t_2$$

And so $\alpha(u, t_1) + (1-\alpha)(v, t_2) = (\alpha u + (1-\alpha) v, \alpha t_1 + (1-\alpha) t_2) \in epi(f) \Rightarrow epi(f)$ is a convex set.
"⇐": Let $epi(f)$ be a convex set.
Let $u, v \in V$ and $\alpha \in [0,1]$.
It holds that $(u, f(u)), (v,) f(v) \in epi(f)$ as $f(x) \leq f(x)$, and so by $epi(f)$ convexity it holds that:

$$\begin{gathered} \alpha(u, f(u)) + (1-\alpha)(v, f(v)) = (\alpha u + (1-\alpha) v, \alpha f(u) + (1-\alpha) f(v)) \in epi(f) \\ \Updownarrow \\ f(\alpha u + (1-\alpha) v) \leq \alpha f(u) + (1-\alpha) f(v) \end{gathered}$$

$\Rightarrow f$ is a convex function. □

(d) Let $f_i : V \to \mathbb{R}$ for any $i \in I$ be a convex function.

Let $\alpha \in [0,1]$ and $u, v \in V$:

$$f(\alpha u + (1-\alpha)v) = \underset{i\in I}{sup} f_i(\alpha u + (1-\alpha)v) \underset{f_i \text{convex \& supremum monotonicity}}{\leq}$$
$$\leq \underset{i\in I}{sup}[\alpha f_i(u) + (1-\alpha) f_i(v)] \underset{\text{sup arithmetics}}{=}$$
$$\alpha \underset{i\in I}{sup} f_i(u) + (1-\alpha) \underset{i\in I}{sup} f_i(v) = \alpha f(u) + (1-\alpha) f(v)$$

□

5. (a) Let $f(w) = l^{hinge}_{x,y}(w,b) = max\left(0, 1 - y\left(w^T x + b\right)\right)$.
As we have seen in recitation and proved above, the maximum of two convex functions is also convex, and as such since 0 is trivialy convex, it is left to show the $1 - y\left(w^T x + b\right)$ is convex in w, b and the rest would follow.
Let us use the second order condition for convexity:

$$\frac{\partial}{\partial w} 1 - y\left(w^T x + b\right) = -yx \Rightarrow \begin{cases} \frac{\partial}{\partial w} - yx = 0 \\ \frac{\partial}{\partial b} - yx = 0 \end{cases}$$
$$\frac{\partial}{\partial b} 1 - y\left(w^T x + b\right) = -y \Rightarrow \begin{cases} \frac{\partial}{\partial w} - y = 0 \\ \frac{\partial}{\partial b} - y = 0 \end{cases}$$

And so the hessian of $1 - y\left(w^T x + b\right)$ is a zeros matrix which is clearly a PSD, and thus $1 - y\left(w^T x + b\right)$ (a linear function) is convex in w, b.
As mentioned $l^{hinge}_{x,y}$ as a maximum of two convex functions is also convex. □

(b) **If** $1 - y\left(w^T x + b\right) \geq 0$: then $l^{hinge}_{x,y}(w,b) = 1 - y\left(w^T x + b\right)$ and as it is differentiable at w, b and it's gradient is $-y(x+1)$ thus it is also a sub-gradient of it.
If $1 - y\left(w^T x + b\right) < 0$: then $l^{hinge}_{x,y}(w,b) = 0$ and as it is differentiable at w, b and it's gradient is 0 thus it is also a sub-gradient of it.

(c) Let $f_1 \ldots, f_m : \mathbb{R}^d \to \mathbb{R}$ convex functions and $\xi_k \in \partial f_k(x)$ for all k, define $f(x) = \sum_{i=1}^m f_i(x)$.
Let $x_0 \in \mathbb{R}^d$ then by the sub-gradient defenition it holds that for any $k \in [m]$ $f_k(x) \geq f_k(x_0) + \langle \xi_k, x - x_0 \rangle$, as such:

$$f(x) = \sum_{i=1}^m f_i(x) \geq \sum_{i=1}^m \left(f_i(x_0) + \langle \xi_i, x - x_0 \rangle\right)$$
$$= \sum_{i=1}^m f_i(x_0) + \sum_{i=1}^m \langle \xi_i, x - x_0 \rangle = f(x_0) + \sum_{i=1}^m \xi_i^T (x - x_0)$$
$$= f(x_0) + \left(\sum_{i=1}^m \xi_i\right)^T (x - x_0) = f(x_0) + \left\langle \sum_{i=1}^m \xi_i, x - x_0 \right\rangle$$

and so by defention, $\sum_{i=1}^m \xi_i \in \partial f(x) = \partial \sum_{i=1}^m f_i(x)$.□

(d) Let $f(w,b) = \frac{1}{m}\sum_{i=1}^m l^{hinge}_{x_i,y_i}(w,b) + \frac{\lambda}{2}||w||^2$ and $w, b \in \mathbb{R}^d$.
Firstly, as seen in recitation $\partial(\alpha f) = \alpha \cdot \partial f$ as for any $g \in \partial h(x)$ then $\alpha h(x) \geq \alpha(h(x_0) + \langle g, x - x_0 \rangle) = \alpha h(x_0) + \langle \alpha g, x - x_0 \rangle$.
Next, as $\frac{\lambda}{2}||w||^2$ is differentiable and convex, using the chain rule, its sub-gradient for any w is $\lambda||w||$.
Lastly, as we have shown above, for finite sum of convex functions any sub-gradients sum is sub-gradient

of the sum of the functions, using the aforementioned in addition to our proof of $l_{x,y}^{hinge}$ being a convex function, we may apply sub-gradient arithmetics:

$$\partial f(w,b) = \partial \frac{1}{m} \sum_{i=1}^{m} l_{x_i,y_i}^{hinge}(w,b) + \frac{\lambda}{2} ||w||^2 =$$
$$\frac{1}{m} \sum_{i=1}^{m} \partial l_{x_i,y_i}^{hinge}(w,b) + \lambda ||w||$$

Therefore setting $\partial l_{x_i,y_i}^{hinge}$ to be the sub-gradient found in section (b) we find a member of the sub-gradient of f for each w.

4 Multi Layer Perceptron (MLP) for for digit classification (MNIST)

7. (a) I have experimented with the following values:

Figure 1: Mini batch size

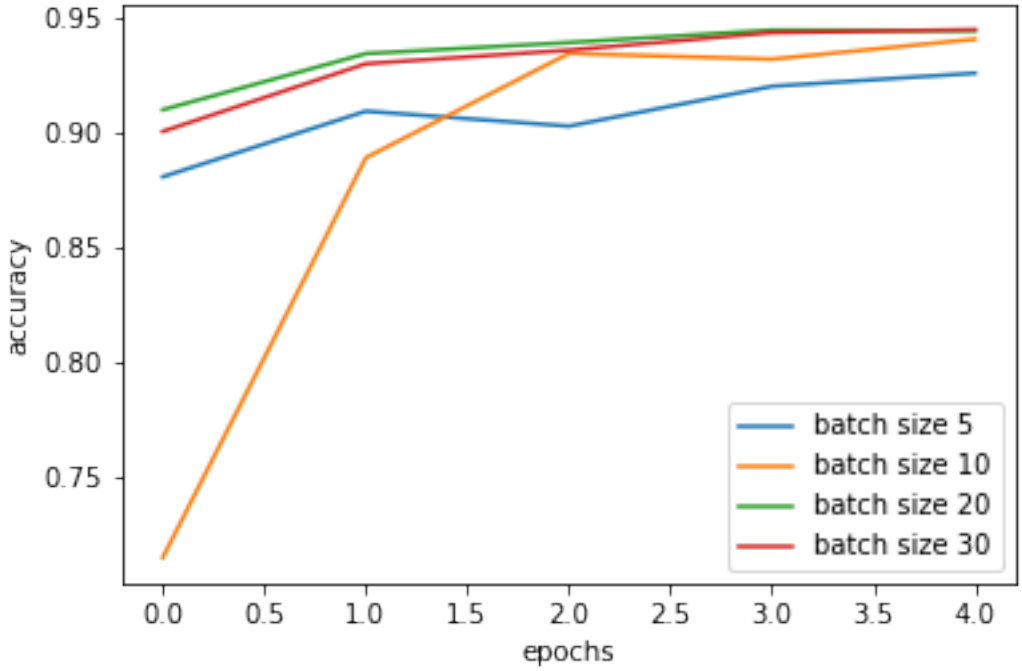


Figure 2: Learning rate

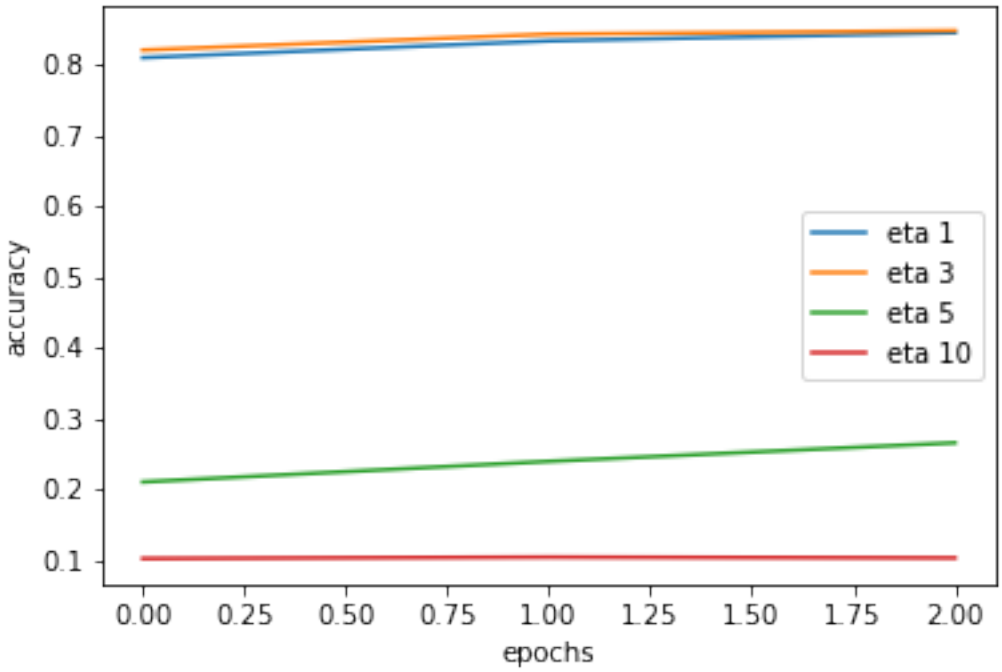
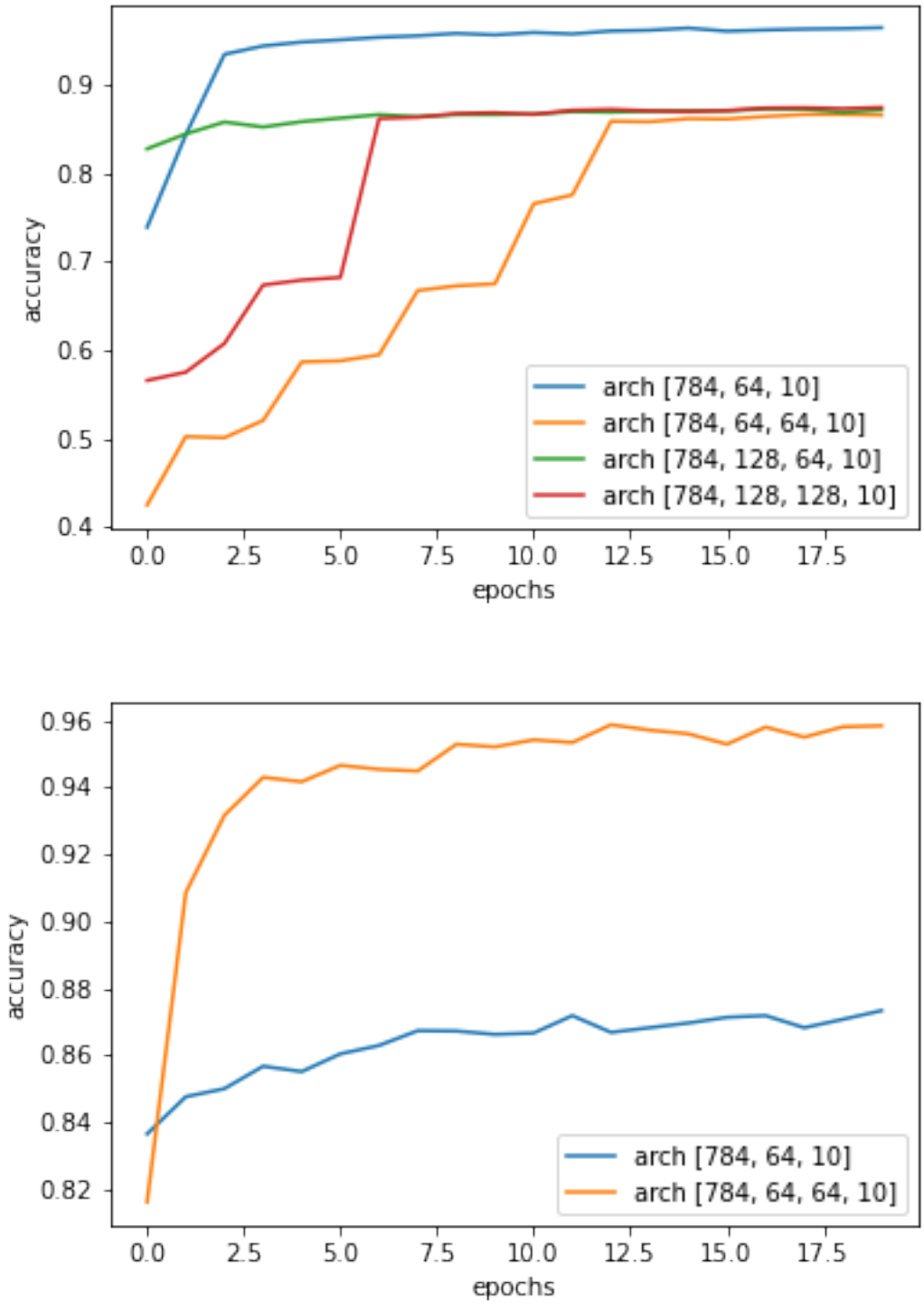


Figure 3: Architecture



(b) I have concluded that the optimal architecture is either $[784, 64, 64, 10]$ with batch size of 20, and learning rate set to 3. It seems by the figures above that the actual architecture of the model had the learning rate has the greatest effect on the module.
The best accuracy I have achieved within 30 epochs was ~0.965.

Figure 4: Final Model