CS229 Fall 2017, Problem Set #3:
Deep Learning & Supervised Learning

Armand Sumo – armandsumo@gmail.com

June 16, 2021

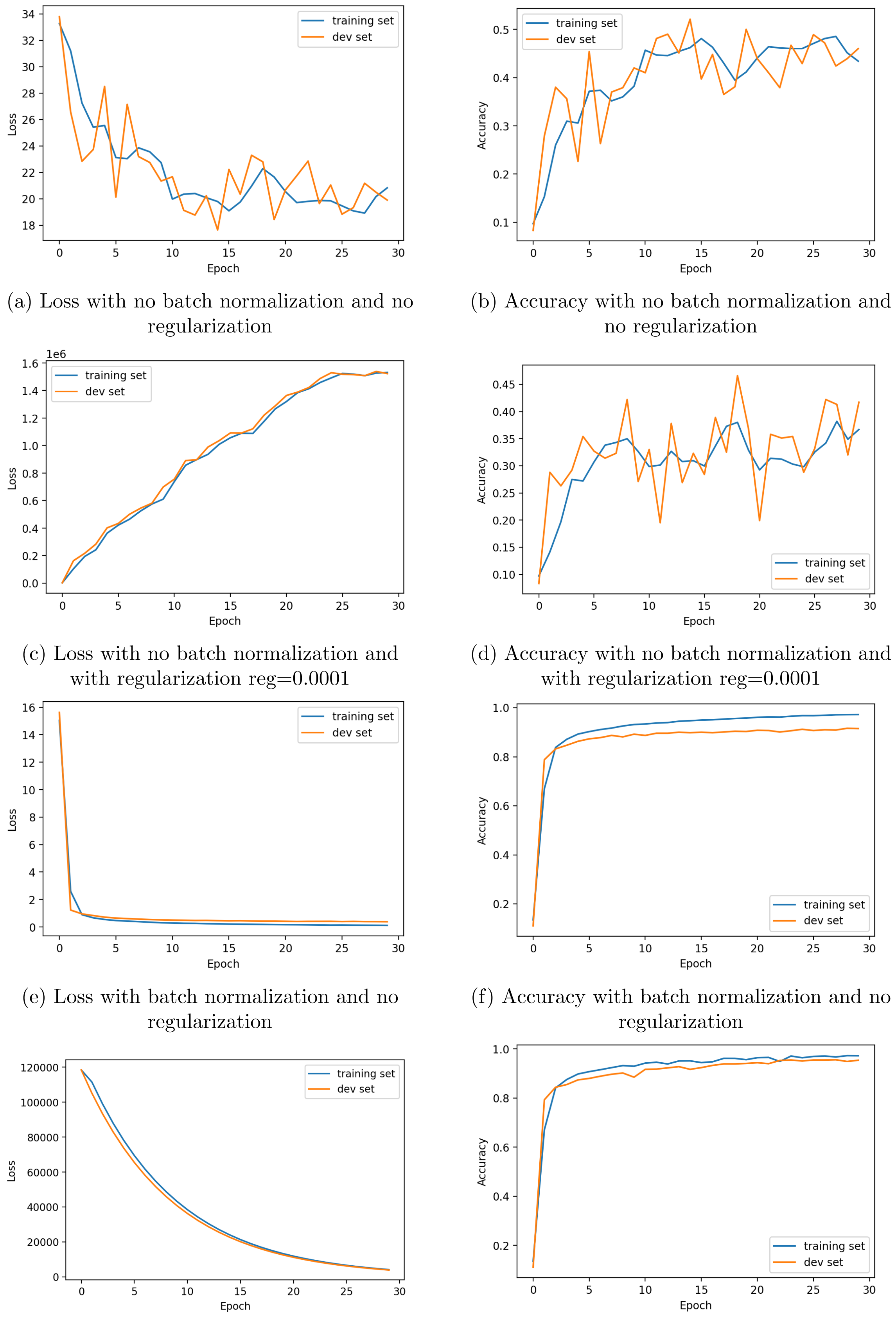
Collaborators:

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

(The code can be found in the files two_layer_net.py, ica.py and rl.py)

1. Neural Networks: MNIST image classification

(a) Adding batch normalization seems to be the most important aspect to increase training accuracy. Regularization does however to reduce over fitting as it reduces the difference in accuracy between the dev and training set. The final accuracy on the training set is 0.986 and on the test set 0.957.



(a) Loss with no batch normalization and no regularization

(b) Accuracy with no batch normalization and no regularization

(c) Loss with no batch normalization and with regularization reg=0.0001

(d) Accuracy with no batch normalization and with regularization reg=0.0001

(e) Loss with batch normalization and no regularization

(f) Accuracy with batch normalization and no regularization

(g) Loss with batch normalization and regularization reg=1

(h) Accuracy with batch normalization and regularization reg=1

Figure 1: Loss and Accuracy with and without regularization and batch normalization on training and dev set

2. EM Convergence

We start from the fact that the EM algorithm converges on some value θ^*. Then,

$$\left(\nabla_\theta \sum_i \mathrm{ELBO}(x^{(i)}, Q_i, \theta)\right)_{\theta=\theta^*} = 0$$

with

$$\begin{aligned}
\mathrm{ELBO}(x, Q, \theta) &= \sum_{z^{(i)}} \log Q(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}, \theta)}{Q(z^{(i)})} && (1)\\
&= \sum_{z^{(i)}} Q(z^{(i)}) \log p(x^{(i)}, z^{(i)}, \theta) - \sum_{z^{(i)}} Q(z^{(i)}) \log Q(z^{(i)}) && (2)
\end{aligned}$$

therefore,

$$\begin{aligned}
&\left(\nabla_\theta \sum_i \sum_{z^{(i)}} \log Q(z^{(i)}) p(x^{(i)}, z^{(i)}, \theta) - \nabla_\theta Q(z^{(i)}) \log Q(z^{(i)})\right)_{\theta=\theta^*} && (3)\\
&= \left(\nabla_\theta \sum_i \sum_{z^{(i)}} \log Q(z^{(i)}) p(x^{(i)}, z^{(i)}, \theta)\right)_{\theta=\theta^*} && (4)\\
&= \left(\nabla_\theta \sum_i \mathbb{E}_{z|x,\theta^*}\Big[p(x^{(i)}, z^{(i)}, \theta)\Big]\right)_{\theta=\theta^*} = 0 && (5)
\end{aligned}$$

Going back to the expression of the log-marginal density:

$$\begin{aligned}
\ell(\theta) &= \sum_i \log p(x^{(i)}, \theta)\\
&= \sum_i \log p(x^{(i)}, z^{(i)}, \theta)\\
&= \sum_i \mathbb{E}_{z|x,\theta^*}\Big[\log p(x^{(i)}, \theta)\Big]
\end{aligned}$$

note that this is true for the expected value over any distribution as long as the term it operates on is a constant random variable.

$$\begin{aligned}\ell(\theta) &= \sum_i \mathbb{E}_{z|x,\theta^*}\left[\log \frac{p(x^{(i)}, z^{(i)}, \theta)}{p(z^{(i)} \mid x^{(i)}, \theta)}\right] \\ &= \mathbb{E}_{z|x,\theta^*}\Bigg[\log p(x^{(i)}, z^{(i)}, \theta)\Bigg] - \mathbb{E}_{z|x,\theta^*}\Bigg[\log p(z^{(i)} \mid x^{(i)}, \theta)\Bigg]\end{aligned}$$

We can now take the gradient of the log-marginal density with respect to θ:

$$\nabla_\theta \ell(\theta) = \nabla_\theta \mathbb{E}_{z|x,\theta^*}\Bigg[\log p(x^{(i)}, z^{(i)}, \theta)\Bigg] - \nabla_\theta \mathbb{E}_{z|x,\theta^*}\Bigg[\log p(z^{(i)} \mid x^{(i)}, \theta)\Bigg]$$

Becauses θ^* maximizes the evidence lower bound ELBO, it must minimize the second term of the RHS of (2) hence:

$$\Bigg(\nabla_\theta \mathbb{E}_{z|x,\theta^*}\Bigg[\log p(z^{(i)} \mid x^{(i)}, \theta)\Bigg]\Bigg)_{\theta=\theta^*} = 0$$

Using (5), we can conclude:

$$\Bigg(\nabla_\theta \ell(\theta)\Bigg)_{\theta=\theta^*} = 0 \tag{6}$$

To eliminate the possibility of θ^* being a saddle point or a minimum of $\ell(\theta)$, we can take the Jacobian of $\nabla_\theta \ell(\theta)$:

$$\begin{aligned}\Bigg(\nabla_\theta^T \nabla_\theta \ell(\theta)\Bigg)_{\theta=\theta^*} &= \Bigg(\nabla_\theta^T \nabla_\theta \mathbb{E}_{z|x,\theta^*}\Bigg[\log p(x^{(i)}, z^{(i)}, \theta)\Bigg]\Bigg)_{\theta=\theta^*} - \\ &\Bigg(\nabla_\theta^T \nabla_\theta \mathbb{E}_{z|x,\theta^*}\Bigg[\log p(z^{(i)} \mid x^{(i)}, \theta)\Bigg]\Bigg)_{\theta=\theta^*}\end{aligned}$$

The first Hessian matrix is positive definite, because θ^* maximizes the expected value of the joint distribution under Q. The second Hessian matrix is negative definite because θ^* minimizes the expected value of the conditional distribution of the latent variable given the observed variable. hence,

$$\Bigg(\nabla_\theta^T \nabla_\theta \ell(\theta)\Bigg)_{\theta=\theta^*} \succ 0$$

This shows that when the EM converges, we have locally maximized the log-marginal of the data.

3. PCA

We begin by expanding the expression the the mean squared error between the projected points and original points:

$$\begin{aligned}
\text{MSE} &= \frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} + f_u(x^{(i)})\right\|_2^2 \\
&= \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - f_u(x^{(i)}))^T(x^{(i)} + f_u(x^{(i)})) \\
&= \frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|_2^2 - 2f_u(x^{(i)})^T x^{(i)} + \left\|f_u(x^{(i)})\right\|_2^2 \qquad (7)
\end{aligned}$$

From here we can exploit the definition of the projection of x onto the direction given by the unit vector u:

$$f_u(x) = ({x^{(i)}}^T u)u \qquad (8)$$

Replacing in (7), we obtain:

$$
\begin{aligned}
\text{MSE} &= \frac{1}{m}\sum_{i=1}^{m} \left\|x^{(i)}\right\|_2^2 - 2({x^{(i)}}^T u)u^T x^{(i)} - \left\|({x^{(i)}}^T u)u\right\|_2^2 \\
&= \frac{1}{m}\sum_{i=1}^{m} \left\|x^{(i)}\right\|_2^2 - 2({x^{(i)}}^T u)u^T x^{(i)} + (({x^{(i)}}^T u)u)^T({x^{(i)}}^T u)u \\
&= \frac{1}{m}\sum_{i=1}^{m} \left\|x^{(i)}\right\|_2^2 - 2({x^{(i)}}^T u)u^T x^{(i)} + u^T(({x^{(i)}}^T u)u^T x^{(i)})u \\
&= \frac{1}{m}\sum_{i=1}^{m} \left\|x^{(i)}\right\|_2^2 - \frac{1}{m}\sum_{i=1}^{m}({x^{(i)}}^T u)u^T x^{(i)} \\
&\qquad\qquad\qquad\qquad\text{(after doing some “transpose surfing”)} \\
&= \frac{1}{m}\sum_{i=1}^{m} \left\|x^{(i)}\right\|_2^2 - u^T\left(\frac{1}{m}\sum_{i=1}^{m} {x^{(i)}}^T x^{(i)}\right)u
\end{aligned}
$$

minimizing this subject to $\|u\|_2 = 1$ is equivalent to maximizing the second term subject to $\|u\|_2 = 1$, which gives the principal eigenvector of $u^T\Sigma u$. Thus, the unit length vector u that minimizes the mean squared error between projected points and original points corresponds to the first principal component of the data.

4. Independent component analysis

(see ica.py)

5. Markov decision processes

(a) We consider a MDP with finite state and action spaces, and discount factor $\gamma < 1$. For any two finite-valued vectors V_1 and V_2,

$$\|B(V_2) - B(V_1)\|_\infty = \max_{s \in S} |B(V_2) - B(V_1)|$$

$$= \max_{s \in S} \left| \gamma \left(\max_{a \in A} \sum_{s' \in S} P_{sa}(s') V_2(s') - \max_{a \in A} \sum_{s' \in S} P_{sa}(s') V_1(s') \right) \right|$$

$$= \gamma \max_{s \in S} \max_{a \in A} \sum_{s' \in S} P_{sa}(s') \Big(V_2(s') - V_1(s') \Big)$$

Because $\sum_{s' \in S} P_{sa}(s') = 1$,

$$\|B(V_2) - B(V_1)\|_\infty \leq \gamma \max_{s \in S} \Big(V_2(s') - V_1(s') \Big) = \gamma \|V_2 - V_1\|_\infty \tag{9}$$

This shows that the Bellman update is a γ-contraction in the max-norm.

(b) Suppose B has two fixed points V_1 and V_2 such that $B(V_1) = V_1$, $B(V_2) = V_2$ and $V_1 \neq V_2$. Then using (9),

$$\frac{\|B(V_2) - B(V_1)\|_\infty}{\|V_2 - V_1\|_\infty} = 1 \leq \gamma$$

This is contradictory because $\gamma > 1$. Hence $V_1 = V_2$: the Bellman operator has a unique fixed point.

6. Reinforcement Learning: The inverted pendulum

(a) It took 43 trials before the algorithm converged. (see control.py)

(b)

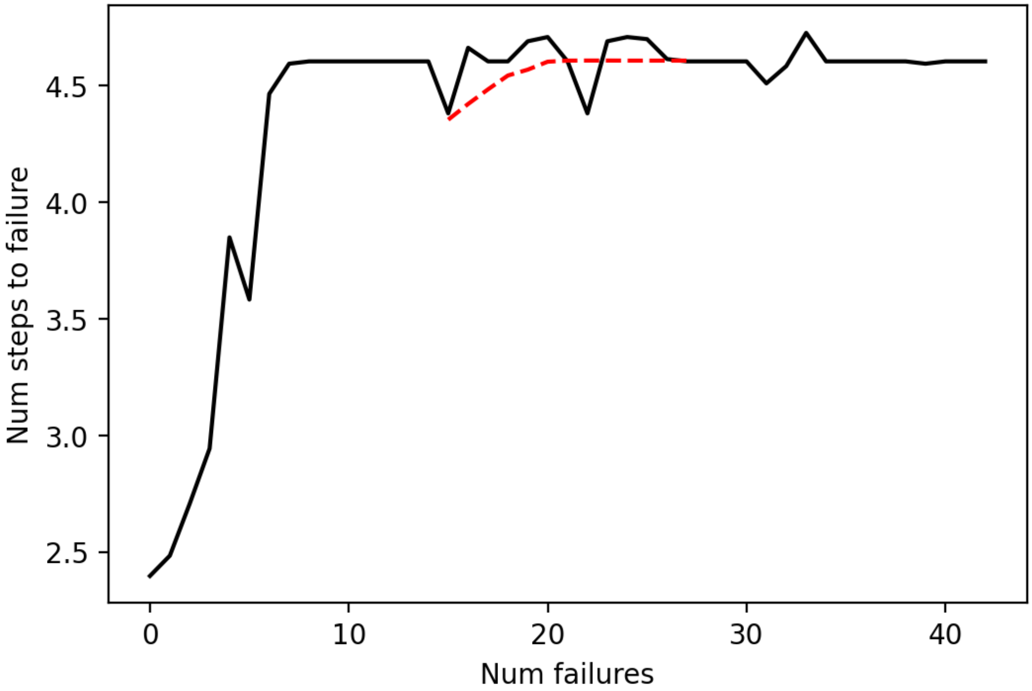


Figure 2: Learning curve showing the number of timesteps the pole was balanced on each trial