Artificial Neural Networks (Gerstner). Exercises for week 7

Error function and Optimization

Exercise 1. Averaging of Stochastic gradients (for ADAM)

We consider stochastic gradient descent in a network with three weights, (w_1, w_2, w_3).

Evaluating the gradient for 100 input patterns (one pattern at a time), we observe the following time series

for w_1: observed gradients are 1.1; 0.9, 1.1; 0.9; 1.1; 0.9; ...

for w_2: observed gradients are 0.1; 0.1; 0.1; 0.1; 0.1; ...

for w_3: observed gradients are 1.1; -0.9; 1.1; -0.9; 1.1; -0.9; ...

a. Calculate the mean gradient (first moment m_1) $\langle g_k \rangle$ for w_k, $k \in [1, 2, 3]$.

b. Calculate the mean of the squared gradient (second moment m_2) $\langle g_k^2 \rangle$ for w_k, $k \in [1, 2, 3]$.

c. Divide the result of (a) by that of (b) so as to calculate $\langle g_k \rangle / \langle g_k^2 \rangle$ as well as $\langle g_k \rangle / \sqrt{\langle g_k^2 \rangle}$ for w_k, $k \in [1, 2, 3]$.

d. The signal-to-noise ratio (SNR) of the gradient g_k is defined as

$$\mathrm{SNR} = \frac{m_1}{\sqrt{\sigma^2}} = \frac{m_1}{\sqrt{m_2 - m_1^2}},$$

where we used the definition of the variance $\sigma^2 = \left\langle (g_k - m_1)^2 \right\rangle = m_2 - m_1^2$ of g_k. In ADAM the weight update is proportional to $\Delta w_k \propto m_1 / \sqrt{m_2}$. With that, show that

- the update in ADAM is proportional to $\Delta w_k \propto \frac{1}{\sqrt{1+1/\mathrm{SNR}^2}} = \frac{\mathrm{SNR}}{\sqrt{\mathrm{SNR}^2+1}}$.
- even though the update in ADAM is not proportional to the SNR, it is proportional to the SNR for small SNR $\to 0$ and saturates for big SNR $\to \infty$.

Exercise 2. Averaging with exponential filters (for ADAM)

In this exercise we study averaging with exponential filters as used e.g. for gradient averaging in SGD with momentum or ADAM.

a. You use an algorithm to update a variable m:

   $m(n+1) = \rho m(n) + (1-\rho) x(n) \qquad (*)$

   where $\rho \in [0, 1)$ and $x(n)$ refers to an observed time series $x(1), x(2), x(3), \ldots$.

   Show that, if all values of x are identical [that is, $x(k) = \bar{x}$ for all k], then the algo (*) converges to $m = \bar{x}$.

b. Assume the initial condition $m(0) = 0$. Show that, for $1 - \rho \ll 1$ the algorithm outputs in time step $n + 1$ the value

$$m(n+1) = (1-\rho) \sum_{k=0}^{n} \exp[-(1-\rho)k] \cdot x(n-k)$$

   Hint: (i) compare $m(n+1)$ with $m(n)$ and reorder terms. (ii) At the end of your calculation you may approximate $\exp[\epsilon] = 1 + \epsilon$ (which is valid for small $\epsilon \ll 1$).

c. Your friend makes the following statement:

The algo (*) performs a running average of the time series $x(n)$ with an exponentially weighted window that extends roughly over $1/(1-\rho)$ samples. Therefore, if you want to include about 100 samples in the average, you should choose $\rho = 0.99$.

Is your friend's claim correct?

Exercise 3. Bias and variance of gradient estimators

For training data $(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^P, y^P)$ and some loss $E(\boldsymbol{w}) = \frac{1}{P}\sum_\mu \ell(f_{\boldsymbol{w}}(\boldsymbol{x}^\mu), y^\mu)$, the gradient is given by $\nabla E(\boldsymbol{w}) = \frac{1}{P}\sum_\mu \nabla\ell(f_{\boldsymbol{w}}(\boldsymbol{x}^\mu), y^\mu)$, with e.g. $\ell(x, y) = \frac{1}{2}(x-y)^2$.

a. In each step of stochastic gradient descent one sample $(\boldsymbol{x}^\mu, y^\mu)$ of the training data is selected. Show that $\nabla\ell(f_{\boldsymbol{w}}(\boldsymbol{x}^\mu), y^\mu)$ is an unbiased estimator of $\nabla E(\boldsymbol{w})$, if each training sample is selected with equal probability. Hint: An estimator $\hat{\theta}$ of a quantity θ is called unbiased, if its expectation $\mathbb{E}\left[\hat{\theta}\right] = \theta$.

b. Instead of single sample stochastic gradient descent it is common practice to use mini-batches. Show that the mini-batch gradient estimator $\frac{1}{M}\sum_{i=1}^M \nabla\ell(f_{\boldsymbol{w}}(\boldsymbol{x}^i), y^i)$, with $1 < M < P$, has lower variance than the single sample estimator, if the samples $(\boldsymbol{x}^i, y^i)$ in each mini-batch are sampled uniformly from the training data.

c. How does this exercise link to Ex. 2 of week 1?

Exercise 4. ADAM and minibatches.

Suppose that in a project you have already spent some time on optimizing the ADAM parameters ρ_1 and ρ_2 while you ran preliminary tests with a minibatch size of 128 on your computer.

For the final run you get access to a bigger and faster computer that allows you to run minibatches of size 512.

How should you rescale ρ_1 and ρ_2 so as to expect roughly the same behavior of the two machines on the training base?

Hint: For ρ_1 you can directly use the results from Exercise 1. However, for ρ_2 you may want to distinguish between the time series for w_1 and that for w_3. Why? Think of the time series in exercise 1 as the gradients of true stochastic gradient. Then make batches of size 2 and 4, and redo the calculation of the squared gradient. What do you observe?

Exercise 5. Unitwise learning rates

Consider minimizing the *narrow valley* function $E(w_1, w_2) = |w_1| + 75|w_2|$ by gradient descent.

a. Sketch the equipotential lines of E, i.e. the points in the $w_1 - w_2$-plane, where $E(w_1, w_2) = c$ for different values of c.

b. Start at the point $\boldsymbol{w}^{(0)} = (10, 10)$ and make a gradient descent step, i.e. $\boldsymbol{w}^{(1)} = \boldsymbol{w}^{(0)} - \eta(\partial E/\partial w_1, \partial E/\partial w_2)$ with $\eta = 0.1$.

   Hint: Use the numeric definition of $\partial|x|/\partial x = sgn(x)$ if $x \neq 0$ and 0 otherwise.

c. Continue gradient descent, i.e .compute $\boldsymbol{w}^{(2)}, \boldsymbol{w}^{(3)}$ and $\boldsymbol{w}^{(4)}$ and draw the points $\boldsymbol{w}^{(0)}, \ldots, \boldsymbol{w}^{(4)}$ in your sketch with the equipotential lines. What do you observe? Can you choose a better value for η such that gradient descent converges faster?

d. Repeat now the gradient descent procedure with different learning rates for the different dimensions, i.e. $\boldsymbol{w}^{(1)} = \boldsymbol{w}^{(0)} - (\eta_1\partial E/\partial w_1, \eta_2\partial E/\partial w_2)$ with $\eta_1 = 1$ and $\eta_2 = 1/75$. What do you observe? Can you choose better values for η_1 and η_2 such that gradient descent converges faster?

e. An alternative to individual learning rates is to use momentum, i.e.
$\Delta \boldsymbol{w}^{(t+1)} = -\eta(\partial E/\partial w_1, \partial E/\partial w_2) + \alpha \Delta \boldsymbol{w}^{(t)}$ with $\alpha \in [0, 1)$ and $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \Delta \boldsymbol{w}^{(t+1)}$.

   Repeat the gradient descent procedure for 3 steps with $\eta = 0.2$ and $\alpha = 0.5$. What do you observe?

f. Assume $\partial E/\partial w_1 = 1$ in all time steps while $\partial E/\partial w_2 = \pm 75$ switches the sign in every time step. Compute $\lim_{t\to\infty} \Delta \boldsymbol{w}^{(t)}$ as a function of η and α. Hint: $\sum_{s=0}^{t} \alpha^s = \frac{1-\alpha^{t+1}}{1-\alpha}$.

g. What do you conclude from this exercise in view of training neural networks by gradient descent with or without momentum?

Exercise 6. Weight space symmetries

Suppose you have found a minimum for some set of weights. Show that in a network with m hidden layers of n neurons each, there are always at least $(n!)^m$ equivalent solutions.

Exercise 7. Relation of weight decay and early stopping

Suppose that we are close to a minimum at w_1^*, w_2^*. The error function in the neighborhood is given by

$$E = \frac{1}{2}\beta_1(w_1 - w_1^*)^2 + \frac{1}{2}\beta_2(w_2 - w_2^*)^2 \tag{1}$$

a. Show that gradient descent with learning rate γ starting at time zero with weights $w_1(0), w_2(0)$ leads to a new weight after n updates given by

   $w_i(n) = w_i^* + (1 - \beta_i\gamma)^n \big(w_i(0) - w_i^*\big)$

b. Suppose that $\beta_2 \gg \beta_1$ (take $\beta_2 = 20\beta_1$). You perform early stopping after n_{stop} steps where $n_{\text{stop}} \approx 1/(5\gamma\beta_1)$.

   Show that at n_{stop} we have $w_2 \approx w_2^*$ and $w_1 \approx w_1(0)$.

   Hint: $\left(1 + \frac{x}{n}\right)^n \approx \exp(x)$ for large n.

   Hence, you may conclude that with an appropriate choice of early stopping, some coordinates have converged and others have not even started convergence.

c. We now consider L2 regularization and work with a modified error function
$\tilde{E} = E + \frac{\lambda}{2}\sum_j (w_j)^2$.

   Show that the minimum of the error function is at

   $w_i = \beta_i w_i^*/(\lambda + \beta_i)$.

d. Consider $\beta_2 \gg \lambda \gg \beta_1$.

   Compare the role of λ with the number n_{stop} in early stopping.

Exercise 8. Simple Perceptron and Bagging

We have four data points:

Two positive examples $t^1 = t^2 = 1$ at $x^1 = (1, 0)^T$ and $x^2 = (0, 1)^T$; and Two negative examples $t^3 = t^4 = 0$ at $x^3 = (0, 0)^T$ and $x^4 = (1, 1)^T$.

a. Draw (with replacement) four times randomly from this data set. What is the probability that you draw each example exactly once?

b. You have generated four new data sets $1 \leq k \leq 4$ by drawing with replacement from the above set. Each set contains four points. You find that in data set k point k is missing $(1 \leq k \leq 4)$.

You work with the perceptron algorithm with hard gain function $g(a) = 1$ for $a > 0$ and zero otherwise.

Make a graph in the data space (input space) and sketch in the graph a solution that the perceptron algorithm finds for data set $k = 1$. Draw the hyperplane.

c. Sketch in the same graph, a solution (one each) that the perceptron algorithm finds for $k = 2, 3, 4$. Label your proposed solutions with $k = 1...4$.

d. Now you perform bagging. What is the value of the (real-valued) bagged output in each region of the above graph in response to an arbitrary data point x^5. In the above graph, give the regions a different texture and write in each region a number which indicates the amplitude of the bagged response.

e. Now you perform majority voting. How many of the 4 data points are correctly classified?

f. Replace the four points by four Gaussian clusters of 25 data points each (Gaussians centered x^1, x^2, x^3, x^4) with standard deviation $\sigma = 0.1$ each; labels are the same for all points inside one Gaussian cluster.) Repeat the above arguments. Assume that the resampled data set k has 20 data points from cluster k, 30 data points from another cluster $k' \neq k$, and 25 from the remaining two clusters.

   Sketch a plot of this new problem on a separate page and repeat the above arguments (draw the hyperplanes etc, parts b - e). Imagine you generate new data points (from the four Gaussians) for the test set. What's the probability for one of those point of not being correctly clustered after bagging with majority vote?