TTIC 31230 Fundamentals of Deep Learning, 2020

Problems For Trainability.

Problem 1. Adjusting Shape for Residual Connections. Consider a computation function f that takes a tensor (such as a layer in a CNN) and returns a tensor of a potentially different shape (such as an image tensor of reduced dimension). We can write a residual network for this case as

$$\begin{aligned} L_1 &= A_1(L_0) + f_1(L_0) \\ L_2 &= A_2(L_1) + f_2(L_1) \\ &\vdots \\ L_N &= A_N(L_{N-1}) + f_N(L_{N-1}) \end{aligned}$$

Here the function A_n adjust the shape of the previous layer to match the shape of $f_n(L_{n-1})$. Suppose that in a CNN we have that $f_n(L_{n+1})$ has the same spatial dimension but fewer features than L_{n-1}.

a. Write a tensor expression for an adjustment function A_n in this case where the dimension is reduced by multiplying the larger dimension feature vector by a matrix.

b. Write a tensor expression for reducing the number of features simply by selecting some features and ignoring others.

c. A standard motivation for residual connections is that there is a direct gradiant path from the loss to the first layer of the network. Which of the adjustment methods of parts I and II preserve this property?

d. Does the shape adjustment given on slide 15 of the notes preserve a direct path from the loss to the first layer?

Problem 2. Improving Trainability with Multiple Loss Heads. Consider a regression problem where we want to predict a scalar value y from a vector x. Consider the L-layer perceptron for this problem defined by the following equations which compute hidden layer vectors $h_1[I], \ldots, h_L[I]$ and predictions $\hat{y}_1, \ldots, \hat{y}_L$ where the prediction $\hat{y}_\ell$ is done with a linear regression on the hidden vector $h_\ell[I]$.

$$
\begin{aligned}
h_0[i] &= x[i] \\
&\vdots \\
h_{\ell+1}[i] &= \sigma(W^{h,h}_{\ell+1}[i,I]h_\ell[I] - B^{h,h}_{\ell+1}[i]) \\
\hat{y}_{\ell+1} &= W^{h,p}_{\ell+1}[I]h_{\ell+1}[I] - B^{h,y}_{\ell+1} \\
&\vdots \\
\text{Loss} &= \sum_{\ell=1}^{L}(y - \hat{y}_\ell)^2
\end{aligned}
$$

Each term $(y - \hat{y}_\ell)^2$ is called a "loss head" and defines a loss on each prediction $\hat{y}_\ell$. Note, however, that there is only one scalar loss minimized by SGD which is the sum of the losses of each loss head.

(a) Explain why these multiple loss terms might improve the ability of SGD to find a useful L-layer MLP regrssion $\hat{y}_L$ when L is large.

(b) As a function of L (ignoring the dimension size I) what is the order of run time for the backpropagation procedure. Explain your answer.

(c) Rewrite the above MLP equations to use residual connections rather than multiple heads. There are multiple correct solutions differing in minor details. Pick one that seems good to you.

Problem 3. Weight Initialization. This problem is on initialization. Consider a single unit defined by

$$y = f(W[I]x[I] - B) = f\left(\left(\sum_i W[i]x[i]\right) - B\right)$$

where B is initialized to zero and f is an activation function such as a sigmoid or ReLU. The vector x is a random variable determined by a random draw of a training example. Assume that the components of x are independent and that each component has zero mean and unit variance. Suppose that we initialize each weight in W from a distribution with zero mean and variance σ^2 and that the distribution is symmetric about zero — (the probability that $w[i] = z$ equals the probability that $w[i] = -z$). For example, $x[i]$ might be distributed as a zero-mean unit-variance Gaussian. Consider $y = \sum_i W[i]x[i]$ as a random variable defined by the distribution on x and the independent random distribution on W. Recall that the variance σ^2 of a sum of independent random variables is the sum of the variances and the variance of a product of zero mean independent random variables is the product of the variances.

(a) What value of σ for $W[i]$ gives zero mean and unit variance for y if the vectors $w[I]$ and $x[I]$ have dimension d? Show your derivation.

(b) For a sigmoid activation function what is the mean of u.

(c) For a sigmoid activation function is the variance of u larger than, equal to, or smaller than the variance of y?

(d) What is the largest possible variance of the output of a sigmoid?

Problem 4. Counting Multiplications in Bottleneck Layers. Consider a bottleneck multi-layer perceptron (MLP) with residual connections defined as follows where N_{bottle} is smaller than $N_{\text{in}} = N_{\text{out}}$.

$$\begin{aligned}
\tilde{L}_\ell[n_{\text{bottle}}] &= \text{ReLU}(W_\ell^{b,1}[n_{\text{bottle}}, N_{\text{in}}]L_\ell[N_{\text{in}}] - B_\ell^{b,1}[n_{\text{bottle}}]) \\
\hat{L}_\ell[n_{\text{out}}] &= \text{ReLU}(W_\ell^{b,2}[n_{\text{out}}, N_{\text{bottle}}]\tilde{L}_\ell[N_{\text{bottle}}] - B_\ell^{b,2}[n_{\text{out}}]) \\
L_{\ell+1}[n] &= L_\ell[n] + \hat{L}_\ell[n]
\end{aligned}$$

(a) What is the number of multiplications done by this network as a function of $N_{\text{in}} = N_{\text{out}} = N$, N_{bottle} and the number of layers L (including the input layer)? Under what conditions does this give fewer multiplications than the standard MLP with one matrix between layers?

(b) We now consider introducing a multiplicative constant γ into the residual connection.

$$L_{\ell+1}[n] = \gamma(L_\ell[n] + \hat{L}_\ell[n])$$

If the network is initialized such that each response of $L_\ell[n]$ and $\hat{L}[n]$ has zero mean and unit variance, and are assummed to be independent, what value of γ gives that $h[\ell + 1, j]$ has zero mean and unit variance.

(c) The main advantage of a stack of residual connections is that there is direct additive path from the loss to each layer of the stack, including the input layer. Give a reason why the introduction of the constant $\gamma < 1$ as in part (b) might be damaging to the optimization of the lower layers of the residual stack.

Problem 5. RNN run time. Consider an autoregressive RNN neural language model with $P_\Phi(w_{t+1}|w_1, \ldots, w_t)$ defined by

$$P_\Phi(w_t|w_1, \ldots, w_{t-1}) = \underset{w_{t+1}}{\text{softmax}} \; e[w_t, I]h[t-1, I]$$

Here $e[w, I]$ is the word vector for word w, $h[t, I]$ is the hidden state vector at time t of a left-to-right RNN, and as described above $e[w, I]h[t, I]$ is the inner prodcut of these two vectors where we have assumed that they have the same dimension. For the first word w_1 we have an externally provided initial hidden state $h[0, I]$ and $w_1, \ldots, w_0$ denotes the empty string. We train the model on

the full loss

$$\begin{aligned}\Phi^* &= \operatorname*{argmin}_{\Phi} E_{w_1,\ldots,w_T\sim\text{Train}} \; -\ln P_\Phi(w_1,\ldots,w_T) \\ &= \operatorname*{argmin}_{\Phi} E_{w_1,\ldots,w_T\sim\text{Train}} \; \sum_{t=1}^{T} -\ln P_\Phi(w_t|w_1,\ldots,w_{t-1})\end{aligned}$$

What is the order of run time as a function of sentence length T for the back-propagation for this model run on a sentence $w_1,\ldots,w_T$? Explain your answer.