Homework 4

1. Architectural choice v.s Vanishing / Exploding Gradients.

As per Piazza @761, this question considers a N layer MLP with a single unit in all the hidden layers and the weight matrices are set to 1.

1.1.1 Effect of Activation - Sigmoid [1pt].

$\sigma(z) = \frac{1}{1+e^{-z}}$.

$\sigma'(z) = \frac{-1}{(1+e^{-z})^2}(-e^{-z})$

$= \frac{e^{-z}}{(1+e^{-z})^2}$

$\max(\sigma'(z)) = 0.25 = \frac{1}{4}$.

$\min(\sigma'(z)) = 0$.

$\left|\frac{\partial f(x)}{\partial x}\right| = \left|\frac{\partial h^n}{\partial x}\right| = \left|\frac{\partial h^n}{\partial h^{n-1}}\frac{\partial h^{n-1}}{\partial x}\right| = \left|\sigma'(h^{n-1})\frac{\partial h^{n-1}}{\partial x}\right|$

$0 \le \left|\frac{\partial f(x)}{\partial x}\right| \le \frac{1}{4}\left|\frac{\partial h^{n-1}}{\partial x}\right|$

$\le \frac{1}{4}^2\left|\frac{\partial h^{n-2}}{\partial x}\right|$

$\le \cdots$

$\le \frac{1}{4}^n$

therefore, $0 \le \left|\frac{\partial f(x)}{\partial x}\right| \le \frac{1}{4}^n$.

As depth n approaches ∞, the gradients will necessarily vanish but not explode.

1.1.2 Effect of Activation - Tanh [1pt]

$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

$\sigma'(z) = 1 - \sigma(z)^2$

$= 1 - \frac{e^{2z} - 2 + e^{-2z}}{e^{2z} + 2 + e^{-2z}}$.

$= \frac{\cancel{e^{2z}} + 2 + \cancel{e^{-2z}} - \cancel{e^{2z}} + 2 - \cancel{e^{-2z}}}{e^{2z} + 2 + e^{-2z}}$

$= \frac{4}{e^{2z} + 2 + e^{-2z}}$

$\max(\sigma'(z)) = 1$

$\min(\sigma'(z)) = 0$

Similarly, $0 \le \left|\frac{\partial f(x)}{\partial x}\right| \le 1\left|\frac{\partial h^{n-1}}{\partial x}\right|$

$\le \cdots$

$\le 1^n = 1$

As depth n approaches ∞, the gradients don't necessarily explode or vanish since it can remain unchanged through backprop.

1.2.1 Gradient through RNN [1pt].

$\sigma_{max}(\frac{\partial x_n}{\partial x_1}) \le \sigma_{max}(\frac{\partial x_n}{\partial x_{n-1}})\sigma_{max}(\frac{\partial x_{n-1}}{\partial x_1})$.

$\le \sigma_{max}(\frac{\partial x_n}{\partial x_{n-1}})\sigma_{max}(\frac{\partial x_{n-1}}{\partial x_{n-2}})\sigma_{max}(\frac{\partial x_{n-2}}{\partial x_1})$

$\le \cdots$

$\le \sigma_{max}(\frac{\partial x_n}{\partial x_{n-1}}) \cdots \sigma_{max}(\frac{\partial x_2}{\partial x_1})$.

$\frac{\partial x_{t+1}}{\partial x_t} = \frac{\partial x_{t+1}}{\partial Wx_t}\frac{\partial Wx_t}{\partial x_t} = diag(\frac{4}{e^{2z_i} + 2 + e^{-2z_i}})W$.

where $z_i = Wx_t$

$\sigma_{max}(\frac{\partial x_{t+1}}{\partial x_t}) \le \sigma_{max}(diag(\frac{4}{e^{2z_i} + 2 + e^{-2z_i}}))\sigma_{max}(W)$

$\propto \sigma_{max}(\cancel{\frac{\partial x_{t+1}}{\partial x_t}})\sigma_{max}(W) = \frac{1}{2}$

Substitute back;

$0 \le \sigma_{max}(\frac{\partial x_n}{\partial x_1}) \le \frac{1}{2}^n$

1.2.3 Benefits of Residual Connections [1pt].

Result from 1.2.2 states that:

$\sigma_{min}(\frac{\partial z_{t+1}}{\partial z_t}) \ge 1 - \sigma_{small}$.

$\sigma_{max}(\frac{\partial z_{t+1}}{\partial z_t}) \ge \sigma_{big} - 1$, $\sigma_{big} >> 2$

So, $\sigma_{max}(\frac{\partial z_{t+1}}{\partial z_t}) >> 1$.

Similarly to 1.2.1,

$\sigma_{max}(\frac{\partial z_n}{\partial z_1}) \le \sigma_{max}(\frac{\partial z_n}{\partial z_{n-1}}) \cdots \sigma_{max}(\frac{\partial z_2}{\partial z_1})$.

will not vanish, since products of terms >> 1 will never be 0.

However, it doesn't solve the exploding gradient problem since the product can still approach ∞.

1.3.2 Batch Normalization and ResNet. [1pt].

the one on the <u>left</u> is easier to learn because its gradient doesn't vanish.

Looking at one such block on the left,

$\frac{\partial h_k}{\partial h_{k-1}} = (\underline{\underline{1}} + \text{grad through block})$.

there will always be a 1 added to the gradient, passing through unchanged, This is not the case for the architecture on the right.

2. Autoregressive Models.

2.2 PixelCNN

2.2.1 Connections [1pt].

$O(WHdk^2)$.

2.2.2 Parallelism [1pt].

$O(d)$

2.3 Multidimensional RNN.

2.3.1 Connections [1pt]

$O(WHdk)$.

Furthermore, in terms of computational complexity, MDRNN is better since it's $O(WHdk)$ # of connections whereas PixelCNN is $O(WHdk^2)$

2.3.3 Discussion [1pt].

PixelCNN is better in terms of parallelization, As discussed before it's sequential operation is $O(d)$, whereas MDRNN neurons' computation are not independent, so they cannot be computed in parallel.