UNIVERSITY
OF WARSAW

Deep Neural Networks - Lecture 3

Marcin Mucha

November 8, 2023

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego Program Operacyjny Polska Cyfrowa na lata 2014-2020, Oś Priorytetowa nr 3 "Cyfrowe kompetencje społeczeństwa" Działanie nr 3.2 "Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej" Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)"

Plan

Saturation

Batch Normalization

Skip connections

Plan

Saturation

Batch Normalization

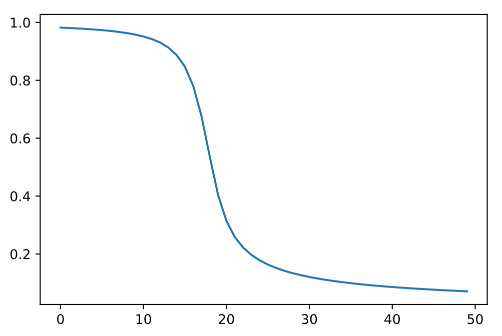
Skip connections

Motivation

Consider a single sigmoid unit trying to learn on a single example $x = 1, y = 0$. The initial weight is $w = 2.0$ and bias is $b = 2.0$ (initial prediction is $\frac{1}{1+e^{-4}} \approx 0.98$). We use a quadratic loss function. This is how loss behaves over 50 epochs.

Motivation

Consider a single sigmoid unit trying to learn on a single example $x = 1, y = 0$. The initial weight is $w = 2.0$ and bias is $b = 2.0$ (initial prediction is $\frac{1}{1+e^{-4}} \approx 0.98$). We use a quadratic loss function. This is how loss behaves over 50 epochs.



This unit is a *saturated* unit.

Motivation

What is the reason for this weird behaviour?

Motivation

What is the reason for this weird behaviour?

$$L(w, b) = (\sigma(w + b))^2 \,.$$

Motivation

What is the reason for this weird behaviour?

$$L(w, b) = (\sigma(w + b))^2 .$$

$$\frac{\partial L(w, b)}{\partial w} = 2\sigma(w + b) \cdot \sigma(w + b) \left(1 - \sigma(w + b)\right) .$$

The second term makes this really small when prediction is close to one - very counter-intuitive.

Log-loss

What happens if we use the log-loss instead?

Log-loss

What happens if we use the log-loss instead?

$$L(w, b) = -\log\left(1 - \sigma(w + b)\right).$$

Log-loss

UNIVERSITY OF WARSAW

What happens if we use the log-loss instead?

$$L(w, b) = -\log\left(1 - \sigma(w + b)\right).$$

$$\frac{\partial L(w, b)}{\partial w} = \frac{1}{1 - \sigma(w + b)} \cdot \sigma(w + b)\left(1 - \sigma(w + b)\right) = \sigma(w + b).$$

Log-loss

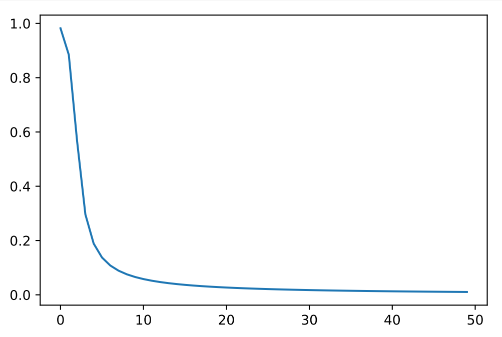
What happens if we use the log-loss instead?

$$L(w, b) = -\log\left(1 - \sigma(w + b)\right).$$

$$\frac{\partial L(w, b)}{\partial w} = \frac{1}{1 - \sigma(w + b)} \cdot \sigma(w + b)\left(1 - \sigma(w + b)\right) = \sigma(w + b).$$

This is perfect!

Log-loss



The message

What does it all mean?

The message

What does it all mean?

Sigmoid (and tanh) units can learn in a very strange way when using GD. But sometimes you can help it.

The message

What does it all mean?

Sigmoid (and tanh) units can learn in a very strange way when using GD. But sometimes you can help it.

When using sigmoid (or tanh) activations in the output layer, L2 loss is a bad idea. Note: loss is often dictated by other considerations.

The message

What does it all mean?

Sigmoid (and tanh) units can learn in a very strange way when using GD. But sometimes you can help it.

When using sigmoid (or tanh) activations in the output layer, L2 loss is a bad idea. Note: loss is often dictated by other considerations.

Real message: when facing difficulties, think about what GD is really doing and figure out the problem!

Exercise

Exercise: Consider the first hidden layer of sigmoid units. Can saturation happen there? Can you avoid it?

Exercise

Exercise: Consider the first hidden layer of sigmoid units. Can saturation happen there? Can you avoid it?

Answer: Of course it can. You can control the initial values in f^l as follows:

- Set biases to 0.
- Rescale inputs so that they are zero-centered and have unit variance.
- Set initial weights to small random values. How small?

Exercise, ctd

Consider input vector $X_1, \ldots, X_m$ and weights $W_1, \ldots, W_m$.

Exercise, ctd

Consider input vector $X_1, \ldots, X_m$ and weights $W_1, \ldots, W_m$.

When $EX_i = 0$ and $EW_i = 0$ and they are independent, we have $\mathrm{Var}(X_i W_i) = \mathrm{Var} X_i \mathrm{Var} W_i = \mathrm{Var} W_i$.

Exercise, ctd

Consider input vector $X_1, \ldots, X_m$ and weights $W_1, \ldots, W_m$.

When $EX_i = 0$ and $EW_i = 0$ and they are independent, we have $\mathrm{Var}(X_i W_i) = \mathrm{Var} X_i \mathrm{Var} W_i = \mathrm{Var} W_i$.

So $\mathrm{Var} \sum (X_i W_i) = \sum \mathrm{Var} W_i$. Obvious solution is to set $\mathrm{Var} W_i = \frac{1}{m}$, or $sd(W_i) = \frac{1}{\sqrt{m}}$.

Exercise, ctd

Consider input vector $X_1, \ldots, X_m$ and weights $W_1, \ldots, W_m$.

When $EX_i = 0$ and $EW_i = 0$ and they are independent, we have $\mathrm{Var}(X_i W_i) = \mathrm{Var} X_i \mathrm{Var} W_i = \mathrm{Var} W_i$.

So $\mathrm{Var} \sum (X_i W_i) = \sum \mathrm{Var} W_i$. Obvious solution is to set $\mathrm{Var} W_i = \frac{1}{m}$, or $sd(W_i) = \frac{1}{\sqrt{m}}$.

The same reasoning can be repeated for all layers, not only input layers. You can also use a similar reasoning for backpropagation step, and get inverse of the number of outputs (Glorot: $\sqrt{\frac{2}{m_i + m_{i+1}}}$).

Sigmoid units?

UNIVERSITY OF WARSAW

Because of saturation problems (but not only), sigmoid units are rarely used nowadays, especially in deep networks. Alternatives: ReLU, max-out and other.

Sigmoid units?

Because of saturation problems (but not only), sigmoid units are rarely used nowadays, especially in deep networks. Alternatives: ReLU, max-out and other.

This does not solve the problem completely - instead of saturation we get dead units etc. Reasoning like this is still very useful.

Plan

Saturation

Batch Normalization

Skip connections

Motivation

Each layer tries to learn a function, but the inputs are changing! - *(internal) covariate shift*.

Motivation

Each layer tries to learn a function, but the inputs are changing! - *(internal) covariate shift*.

Idea: Normalize mean and variance of f^l.

Motivation

Each layer tries to learn a function, but the inputs are changing! - *(internal) covariate shift*.

Idea: Normalize mean and variance of f^l.

Explanation: f^l as combinations of many factors are likely to resemble Gaussians (CLT). By normalizing we keep the distribution of inputs to g^l stable.

Motivation

Each layer tries to learn a function, but the inputs are changing! - *(internal) covariate shift*.

Idea: Normalize mean and variance of f^l.

Explanation: f^l as combinations of many factors are likely to resemble Gaussians (CLT). By normalizing we keep the distribution of inputs to g^l stable.

Exercise: Why it makes less sense to normalize g^l? Give a handwaving argument.

Motivation

Each layer tries to learn a function, but the inputs are changing! - *(internal) covariate shift*.

Idea: Normalize mean and variance of f^l.

Explanation: f^l as combinations of many factors are likely to resemble Gaussians (CLT). By normalizing we keep the distribution of inputs to g^l stable.

Exercise: Why it makes less sense to normalize g^l? Give a handwaving argument.

Answer: Outputs of g^l can have rather complicated distributions. Scaling and shifting might not guarantee stable distributions. That said, recent papers actually advocate this. Apparently it often actually works better!

Implementation

UNIVERSITY OF WARSAW

First idea might be to manually normalize f^l outside the backpropagation engine.

Implementation

First idea might be to manually normalize f^l outside the backpropagation engine.

No longer clear what is really happening inside.
Instead...

Solution

Introduce an intermediate computation between f^l and g^l in our graph, denoted BN^l.

Solution

Introduce an intermediate computation between f^l and g^l in our graph, denoted BN^l.

Consider a layer l and a mini-batch of size s.

$$\mu_i^l = \frac{1}{s} \sum_{j=1}^{s} f_{i,j}^l.$$

So, μ^l is a vector of mini-batch averages for each linearity.

Solution

Introduce an intermediate computation between f^l and g^l in our graph, denoted BN^l.

Consider a layer l and a mini-batch of size s.

$$\mu_i^l = \frac{1}{s}\sum_{j=1}^{s} f_{i,j}^l.$$

So, μ^l is a vector of mini-batch averages for each linearity.

$$v_i^l = \frac{1}{s}\sum_{j=1}^{s}(f_{i,j}^l - \mu_i^l)^2.$$

This is a vector of mini-batch estimates of variances of each f_i^l.

Solution, ctd.

Can we set $BN^l = \frac{f^l - \mu^l}{\sqrt{v^l + \varepsilon}}$ and activate g^l on BN^l?

Solution, ctd.

Can we set $BN^l = \frac{f^l - \mu^l}{\sqrt{v^l + \varepsilon}}$ and activate g^l on BN^l?

This is bad since then the non-linearities will receive restricted inputs. Instead $BN^l = \gamma^l \frac{f^l - \mu^l}{\sqrt{v^l + \varepsilon}} + \beta^l$ and we activate g^l on these BN^l. Note that each node has its own γ and β - learned parameters.

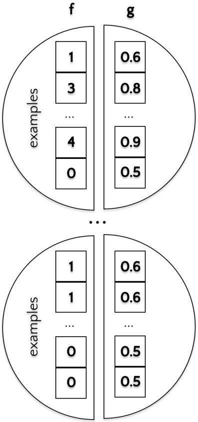
Solution, ctd.

Can we set $BN^l = \frac{f^l - \mu^l}{\sqrt{v^l + \varepsilon}}$ and activate g^l on BN^l?

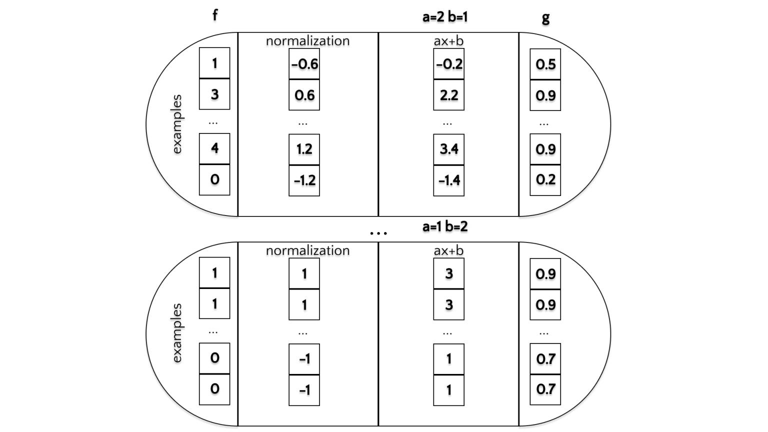
This is bad since then the non-linearities will receive restricted inputs. Instead $BN^l = \gamma^l \frac{f^l - \mu^l}{\sqrt{v^l + \varepsilon}} + \beta^l$ and we activate g^l on these BN^l. Note that each node has its own γ and β - learned parameters.

This is a significantly more complicated computation: each $BN^l_{i,j}$ depends on all $f^l_{i,j}$. But can still compute gradients of L over g^l, BN^l and f^l, and consequently over w's, b's, γ's and β's.

Batch normalization - example



Batch normalization - example



Extra details

When using batch normalization, we do not need biases, they get removed anyway. β's acts as biases.

Extra details

UNIVERSITY OF WARSAW

When using batch normalization, we do not need biases, they get removed anyway. β's acts as biases.

When performing prediction, one might consider computing global mean and variance for the whole dataset (or at least better estimates than a single batch) and using them instead of batch statistics. Unfortunately this introduces a discrepancy between training and testing (same as dropout).

Advantages

Advantages

- Speeds up learning.

Advantages

- Speeds up learning.
- Allows for higher learning rates (more stable gradients).

Advantages

- Speeds up learning.
- Allows for higher learning rates (more stable gradients).
- Acts as a regularizer (can even be used as the only regularizer).

Advantages

UNIVERSITY OF WARSAW

- Speeds up learning.
- Allows for higher learning rates (more stable gradients).
- Acts as a regularizer (can even be used as the only regularizer).
- Avoids saturation for sigmoid and tanh units and dying ReLU units. This makes training deeper networks possible.

Advantages

- Speeds up learning.
- Allows for higher learning rates (more stable gradients).
- Acts as a regularizer (can even be used as the only regularizer).
- Avoids saturation for sigmoid and tanh units and dying ReLU units. This makes training deeper networks possible.
- No careful initialization required.

Recent research I

Laarhoven [2017] argues that:

- If using batch normalization, using L2 regularization as well might be crucial for convergence, BUT
- The L2 regularization coefficient only influences effective learning rate!

Recent research II

Later, Santurkar et al. [2018] argue that:

- Batch normalization helping with internal covariate shift is a myth.
- What batch normalization really does: it smoothens the loss funcion and its gradient.

Recent research III

In a more recent research paper, Brock et al. [2021] manage to obtain SOTA results on ImageNet recognition without batch normalization (they use clipping instead), and almost 10x faster.

Is this a beginning of a revolution? Does not seem so thus far.

Plan

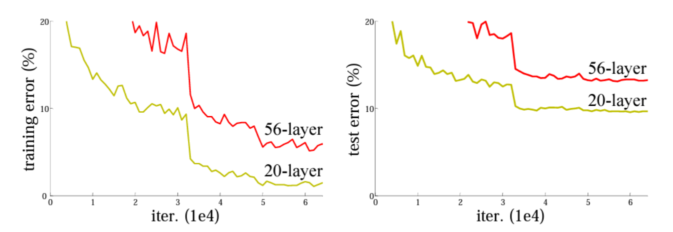
Saturation

Batch Normalization

Skip connections

Deeper networks

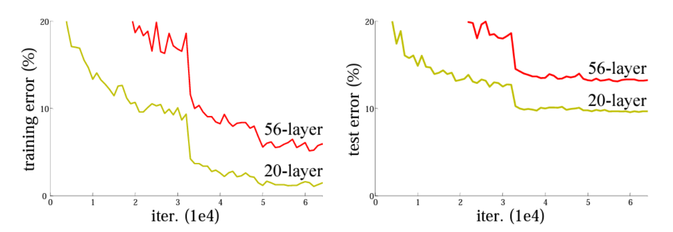
- Deeper network should in theory work better (on the train set) since it can just use identity mapping on extra layers.
- Deeper networks do work better on both train and test in practice (using BN). To a point...



What seems to be the problem?

Deeper networks

- Deeper network should in theory work better (on the train set) since it can just use identity mapping on extra layers.
- Deeper networks do work better on both train and test in practice (using BN). To a point...



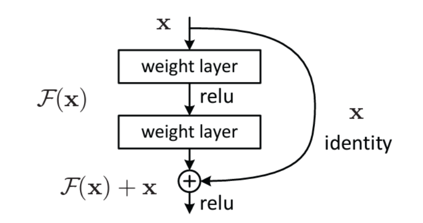
What seems to be the problem?

Identity seems to be a rather difficult mapping to learn!

Residual connections

Solution: residual connection.

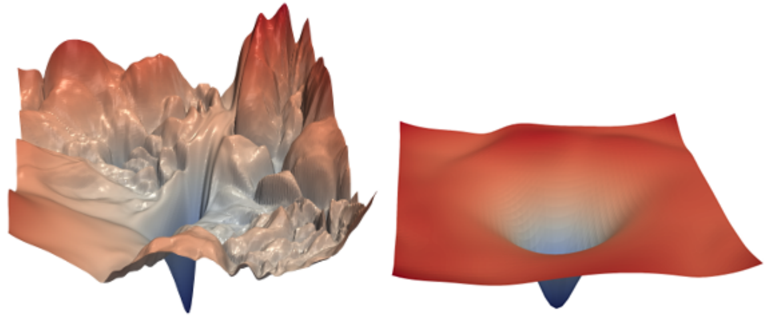
- Signal is copied several layers forward.
- It is ADDED to layers' output, so it learns the RESIDUAL.
- To go deeper we learn the 0 mapping, not identity - easier.
- Dimension needs to agree.
- No new parameters!



Does it work?

Spectacular success stories, e.g. ResNet (next lecture) or transformers.

Today: a suggestive image (for ResNet).



(a) without skip connections (b) with skip connections

Concatenation

Sometimes it makes more sense to use a skip connection to APPEND an early layer to deeper layer.

Useful for “annotating the input” style problems with dense outputs, like image segmentation, etc. (see U-Net in next lecture). Even for shallow networks

Typically these are called “skip connections” (vs “residual connections”), but not always!

Extra reading

Original batch norm paper

Batch norm with L2 regularization

Exploration of what batch norm does

Original ResNet paper

Visualization of skip connection landscape

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego Program Operacyjny Polska Cyfrowa na lata 2014-2020, Oś Priorytetowa nr 3 "Cyfrowe kompetencje społeczeństwa" Działanie nr 3.2 "Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej" Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”