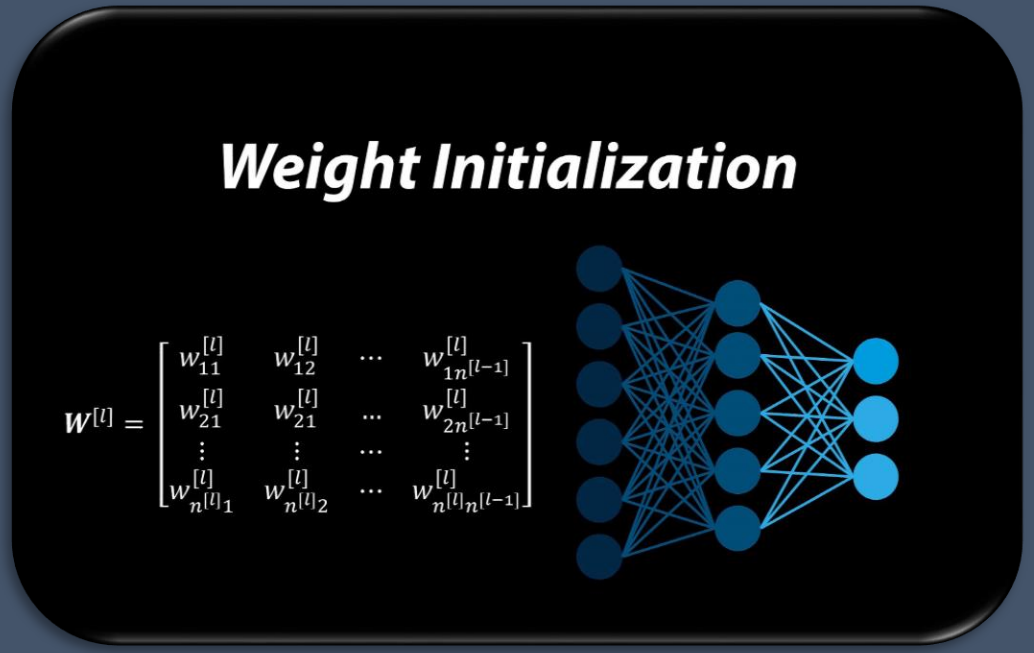
دوره‌ی آموزشی «علم داده»
Data Science Course



جلسه سی‌ام (بخش اول)

مقداردهی اولیه به وزن‌ها

Initialization

مدرس: محمد فزونی
عضو هیئت علمی دانشگاه گنبدکاووس

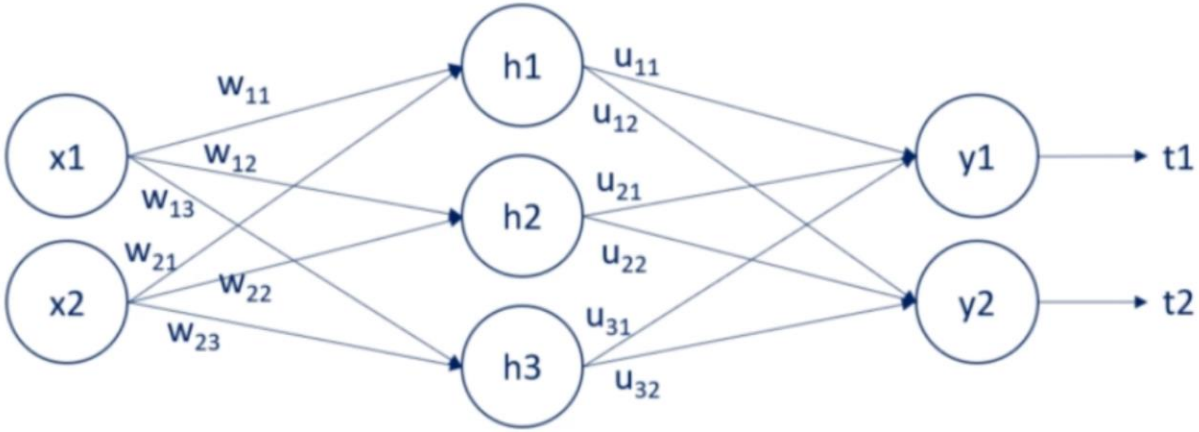
INITIALIZATION

فرایندی که مقادیر اولیه‌ای رو به وزن‌ها تخصیص می‌ده

Initialization is the process in which we set the initial values of weights.

Initialization

تخصیص بد وزن‌ها چه عواقبی دارد؟



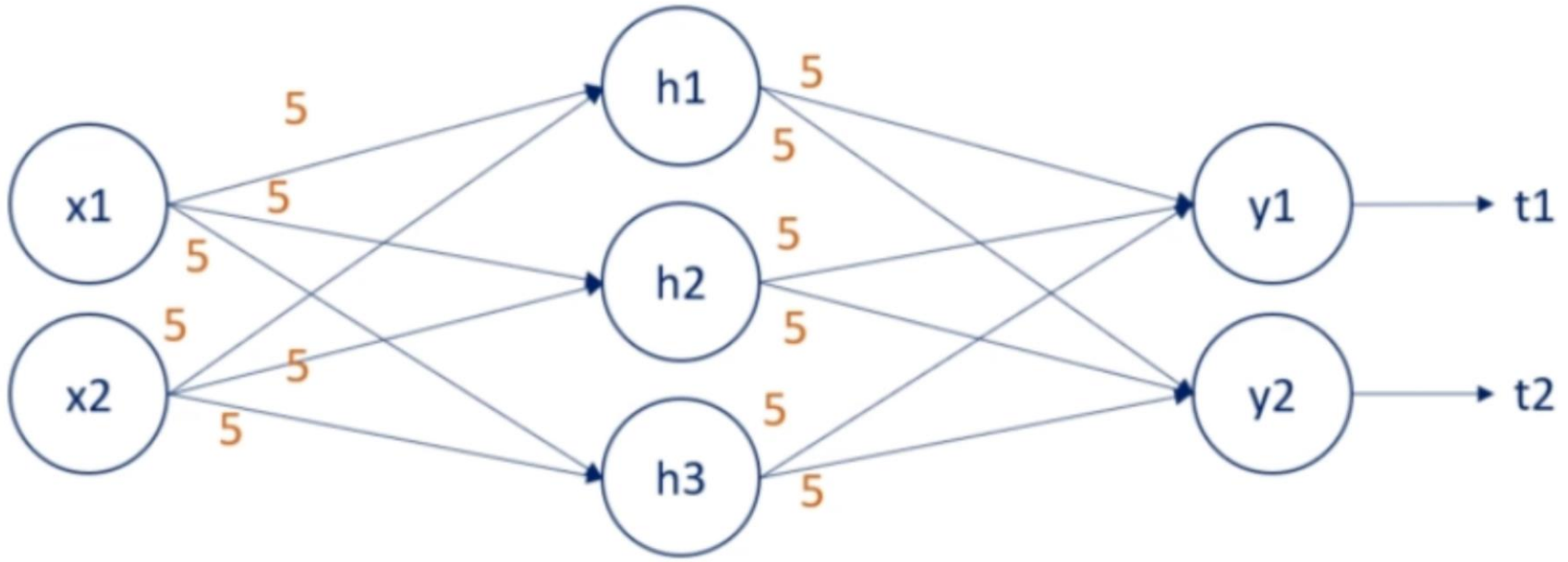
Let's initialize our weights and biases in such a way that they **are all equal to a constant**

اگر فرض کنیم که تمامی وزن‌ها برابر با صفر یا یه عدد ثابت باشند، مدل ما چیزی یاد نمی‌گیره. آپدیت صورت می‌گیره، ولی چندان ارزشی نداره

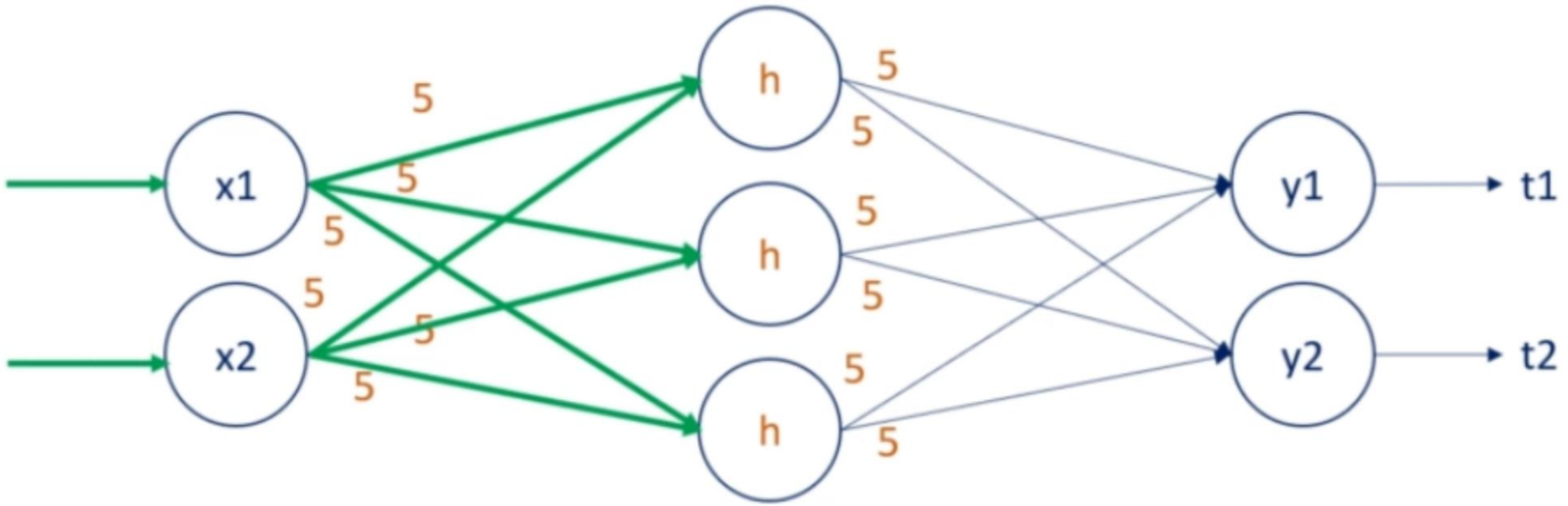
Zero initialization:

If all the weights are initialized to zeros, the derivatives will remain same for every weight. As a result, neurons will learn same features in each iterations. This problem is known as network failing to break symmetry. And not only zero, any constant initialization will produce a poor result.

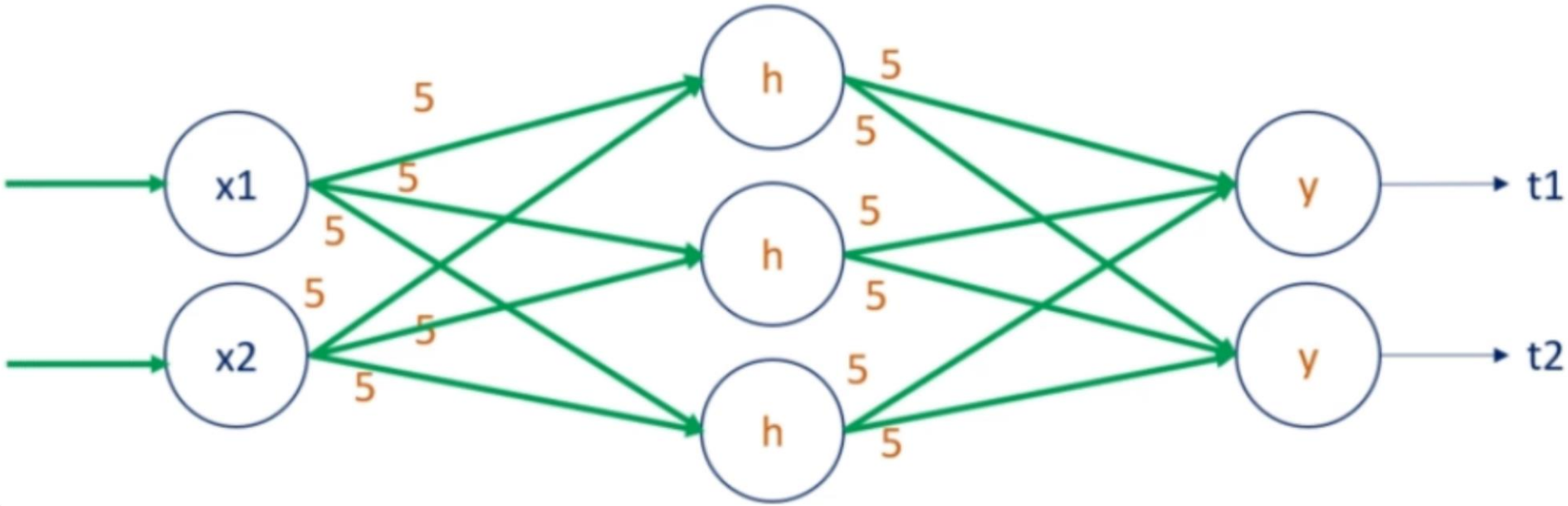
Initialization



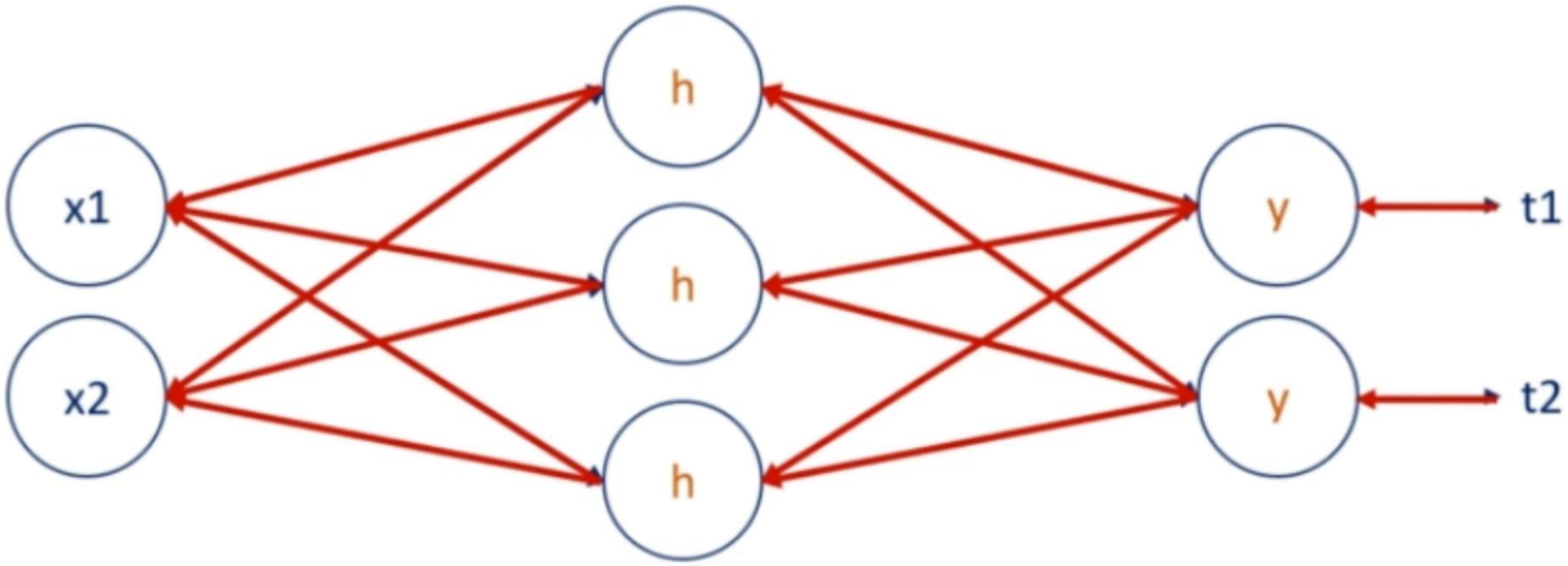
Initialization



Initialization

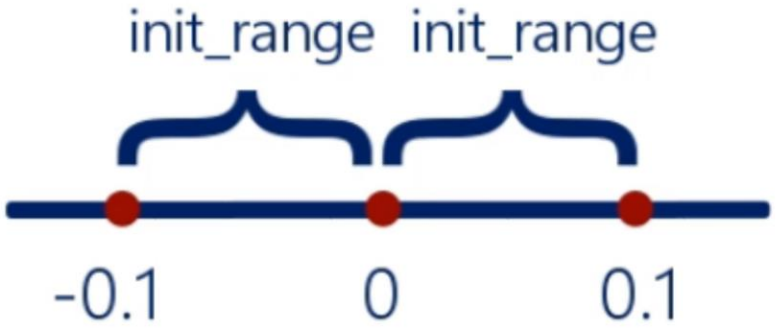


Initialization



انتخاب وزن‌ها به شیوه‌ی تصادفی
(یکنواخت و نرمال)

Our initial weights and biases will be picked randomly from the interval [-0.1,0.1]



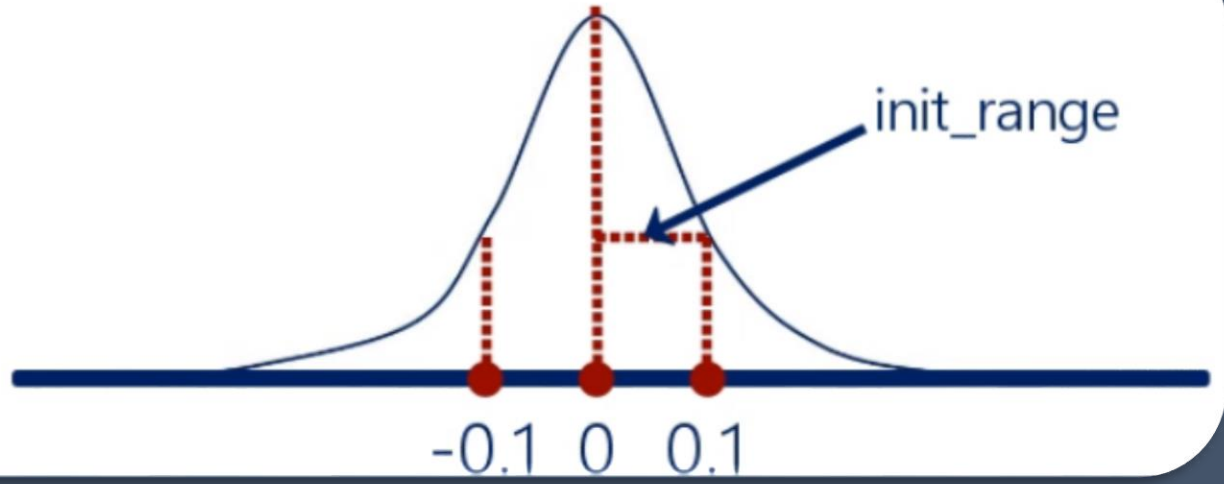
Initialize variables

In [10]:
```
init_range = 0.1

weights = np.random.uniform(-init_range,init_range, size=(2,1))

biases = np.random.uniform(-init_range,init_range,size=1)

print(weights)
print(biases)
```

W
```
[[-0.02758852]
 [-0.03556078]]
```
b
```
[-0.05488636]
```

Each value has an equal probability of being selected

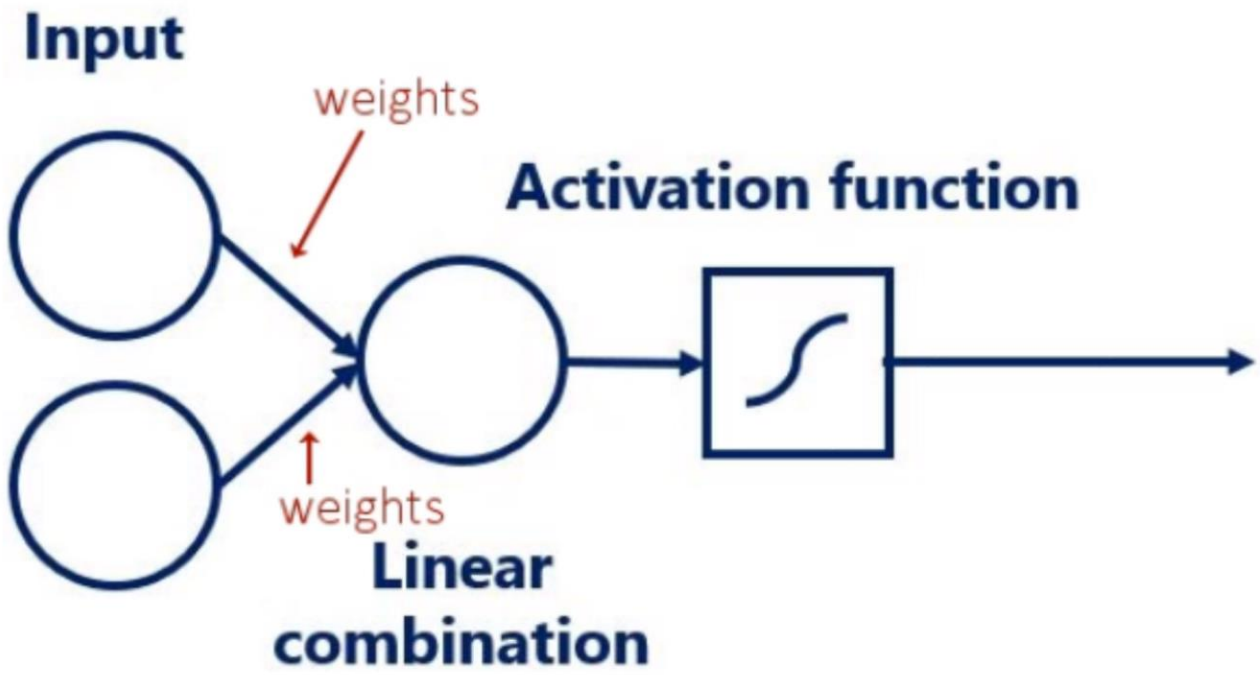
Our initial weights and biases will be picked randomly from the interval [-0.1,0.1] in a random normal manner, where the mean is 0 and the standard deviation is 0.1 (variance 0.01).



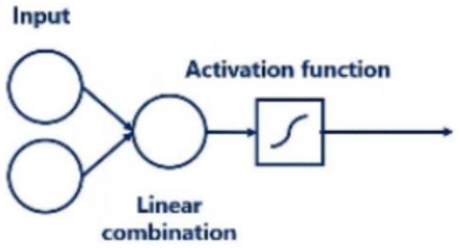
Studies have shown that initializing the weights with values sampled from a random distribution instead of constant values like zeros and ones actually helps a neural net train better and faster.

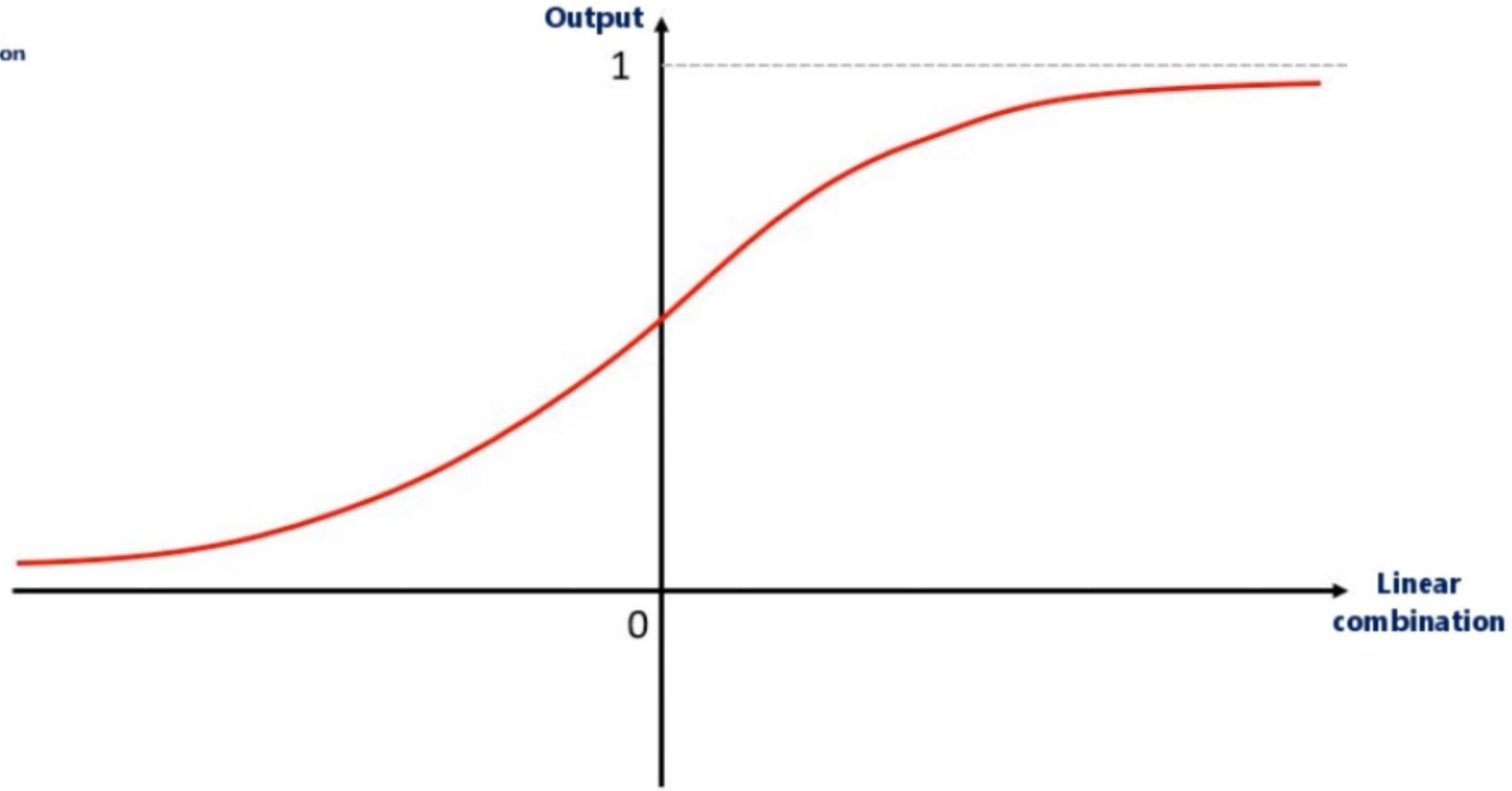
Both methods are somewhat problematic

Problem

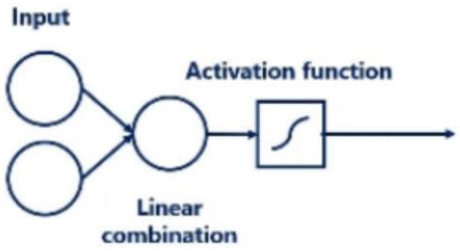

Input
weights
Activation function
weights
Linear
combination

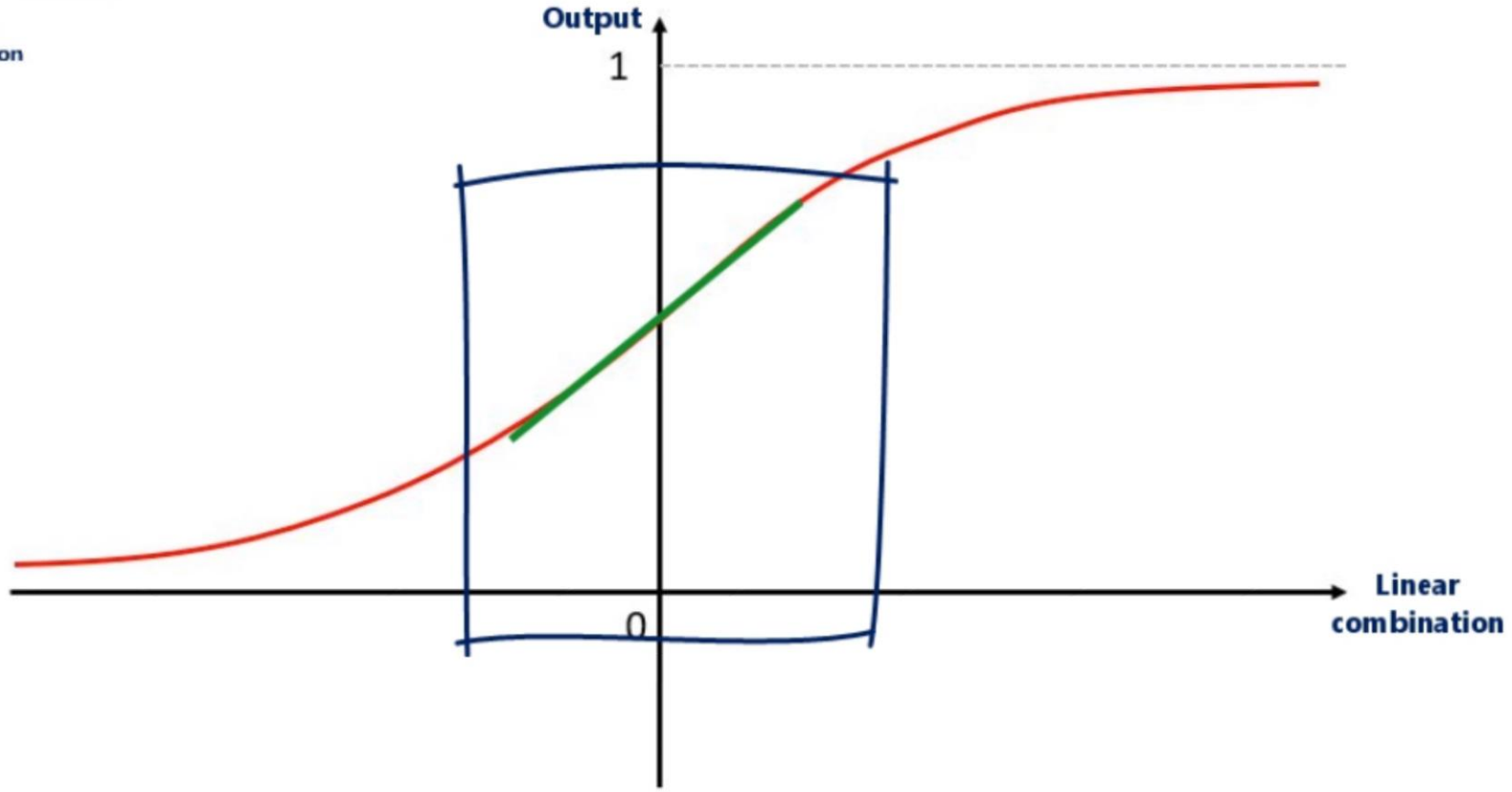
Sigmoid activation function



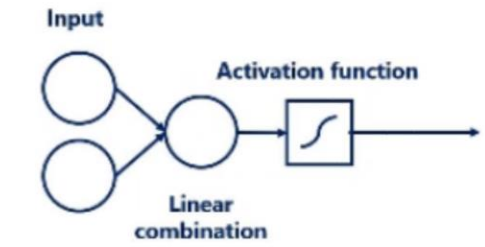


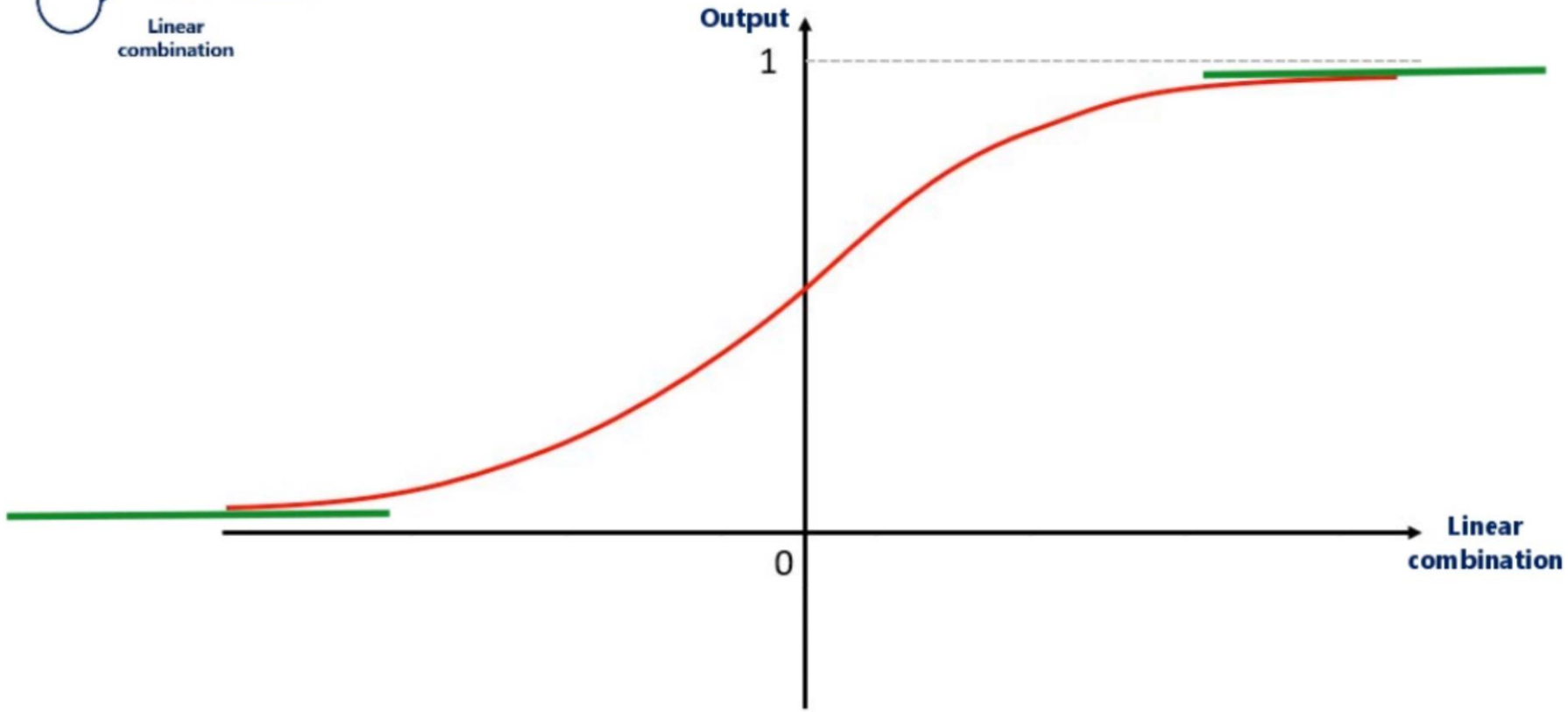
Sigmoid activation function



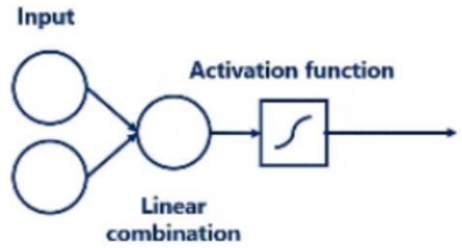


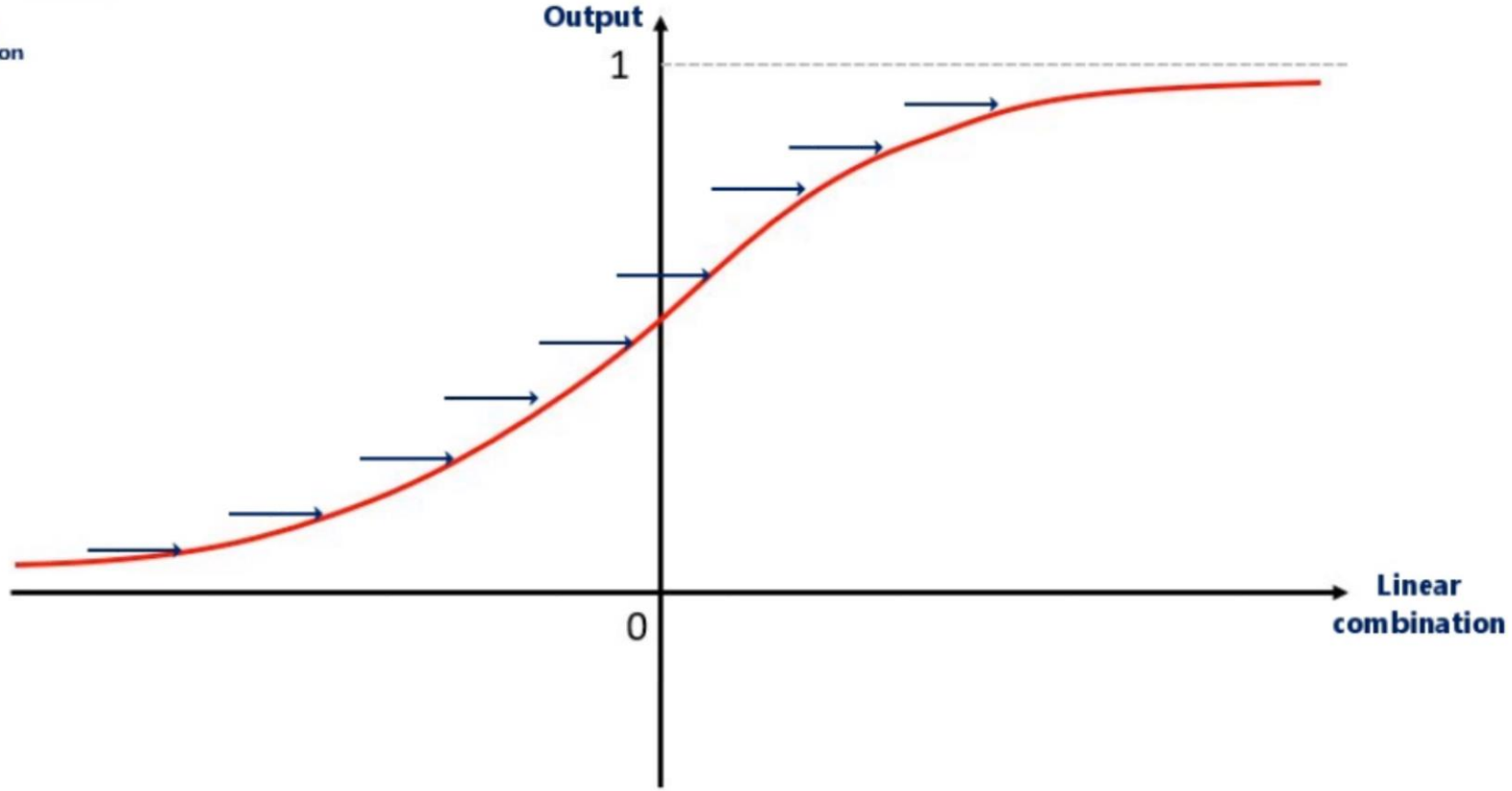
Sigmoid activation function





Sigmoid activation function





Xavier initialization

Glorot initialization

Xavier Glorot

initialization

Xavier initialization

Original paper

Understanding the difficulty of training deep feedforward neural networks

Xavier Glorot, Yoshua Bengio ;
Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR 9:249-256, 2010.

Understanding the difficulty of training deep feedforward neural networks

Xavier Glorot **Yoshua Bengio**
DIRO, Université de Montréal, Montréal, Québec, Canada

Abstract

Whereas before 2006 it appears that deep multi-layer neural networks were not successfully trained, since then several algorithms have been shown to successfully train them, with experimental results showing the superiority of deeper vs less deep architectures. All these experimen-

learning methods for a wide array of *deep architectures*, including neural networks with many hidden layers (Vincent et al., 2008) and graphical models with many levels of hidden variables (Hinton et al., 2006), among others (Zhu et al., 2009; Weston et al., 2008). Much attention has recently been devoted to them (see (Bengio, 2009) for a review), because of their theoretical appeal, inspiration from biology and human cognition, and because of empirical success in vision (Ranzato et al., 2007; Larochelle et al.

Xavier initialization

Uniform Xavier initialization:

Main idea: Method is not so important. The number of **inputs and outputs** is

Normal Xavier initialization:

Xavier initialization

Uniform Xavier initialization: draw each weight, w, from a random uniform distribution

in [-x,x] for $x = \sqrt{\frac{6}{inputs + outputs}}$

Normal Xavier initialization:

Xavier initialization

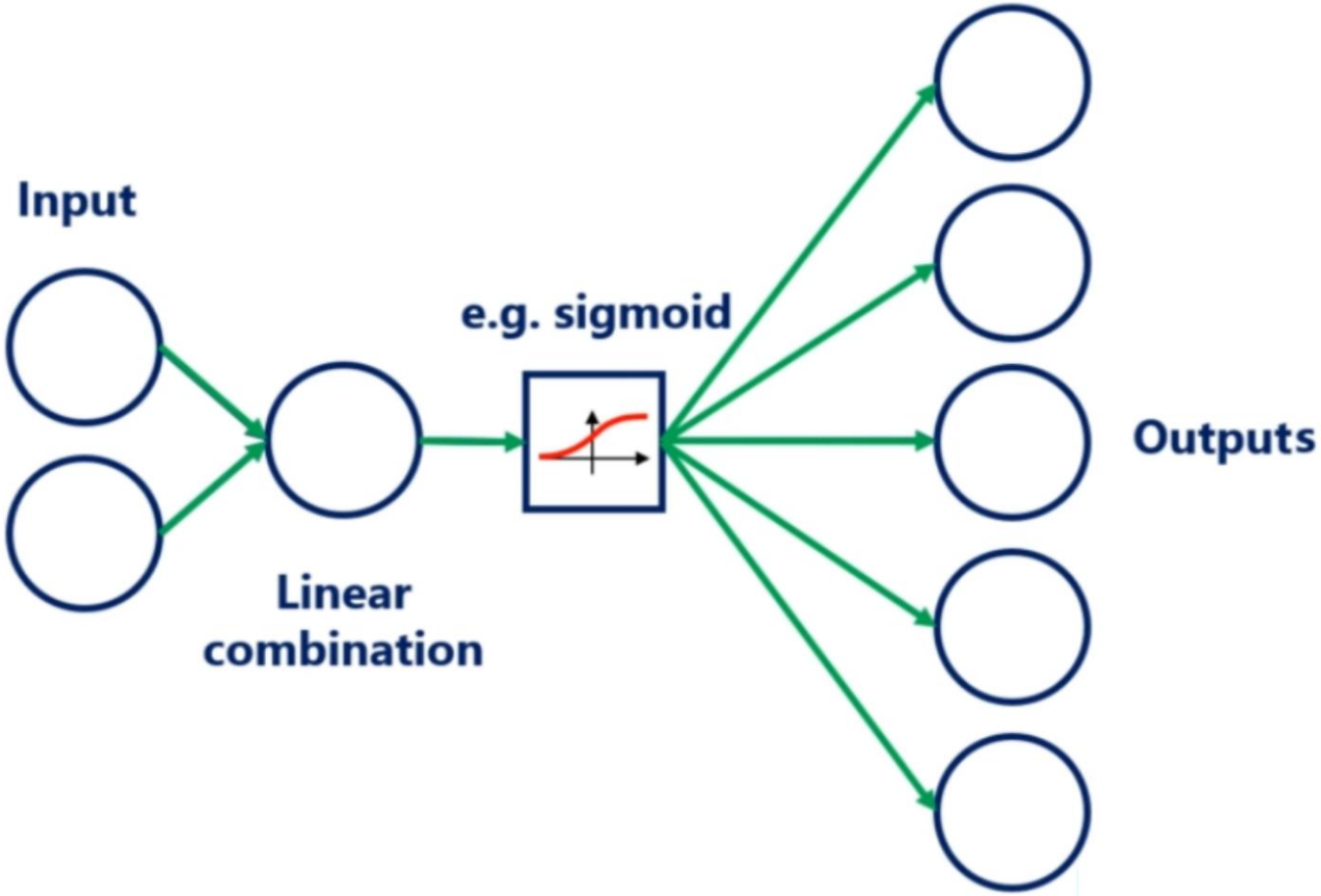
Uniform Xavier initialization: draw each weight, w, from a random uniform distribution

in [-x,x] for $x = \sqrt{\frac{6}{inputs + outputs}}$

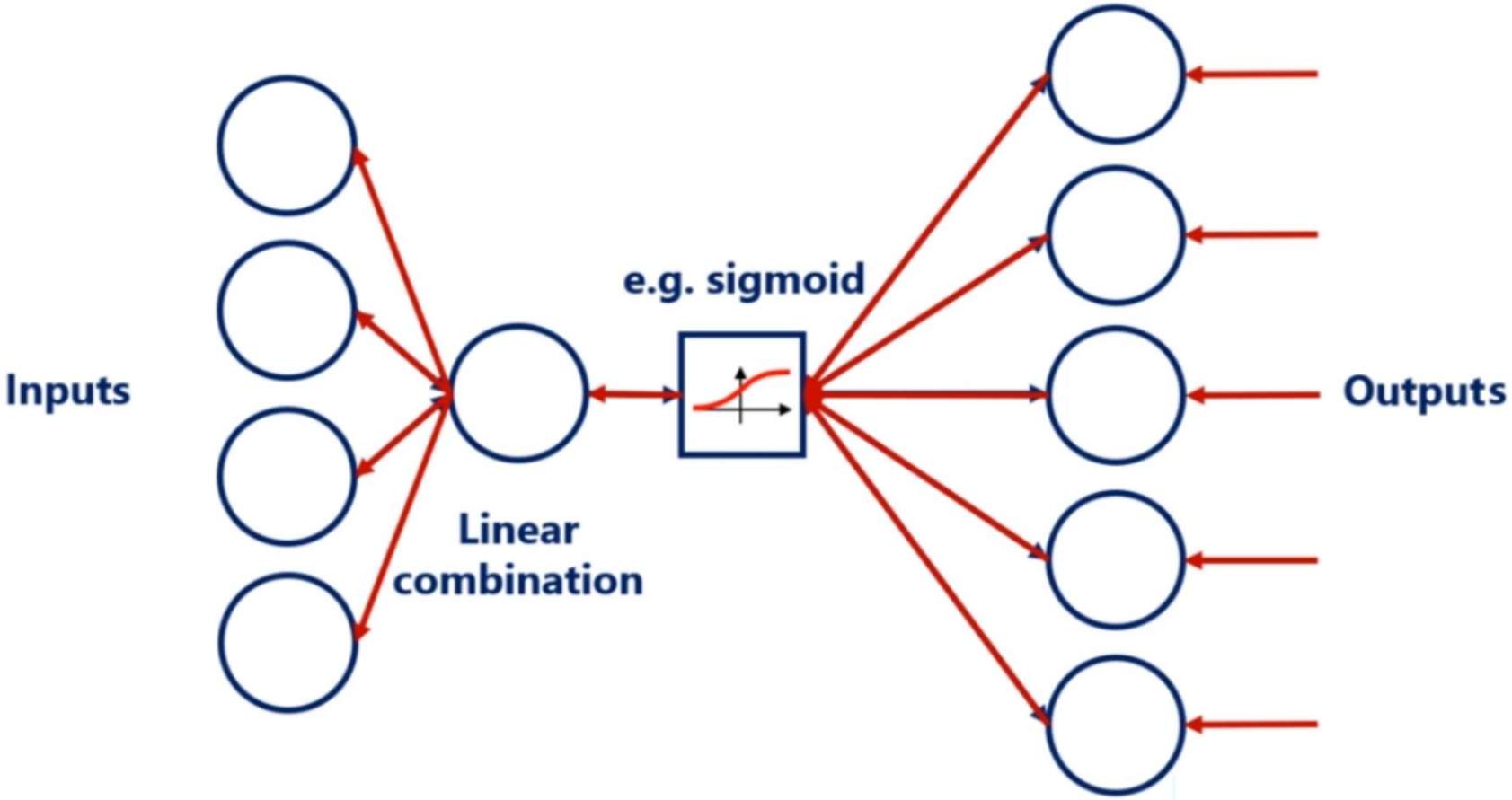
Normal Xavier initialization: draw each weight, w, from a normal distribution with a mean

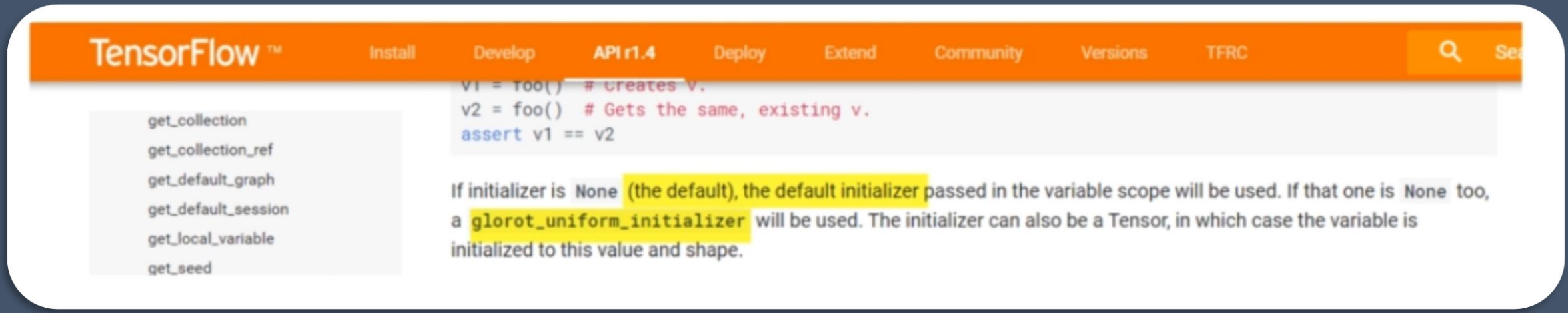
of 0, and a standard deviation $\sigma = \sqrt{\frac{2}{inputs + outputs}}$

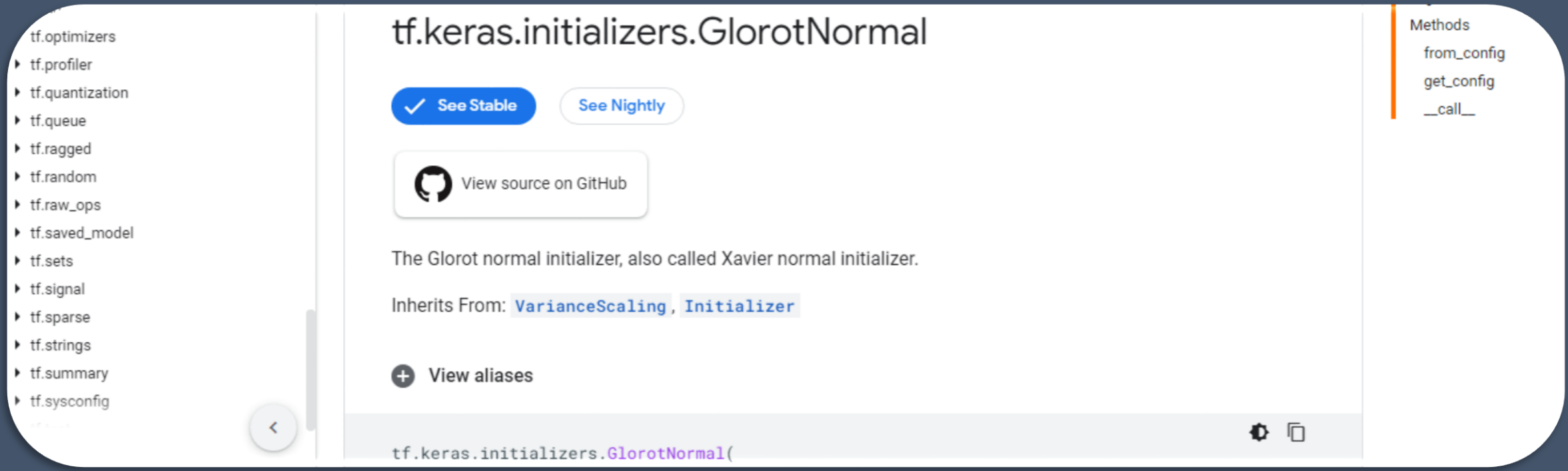
Why do inputs and outputs matter?



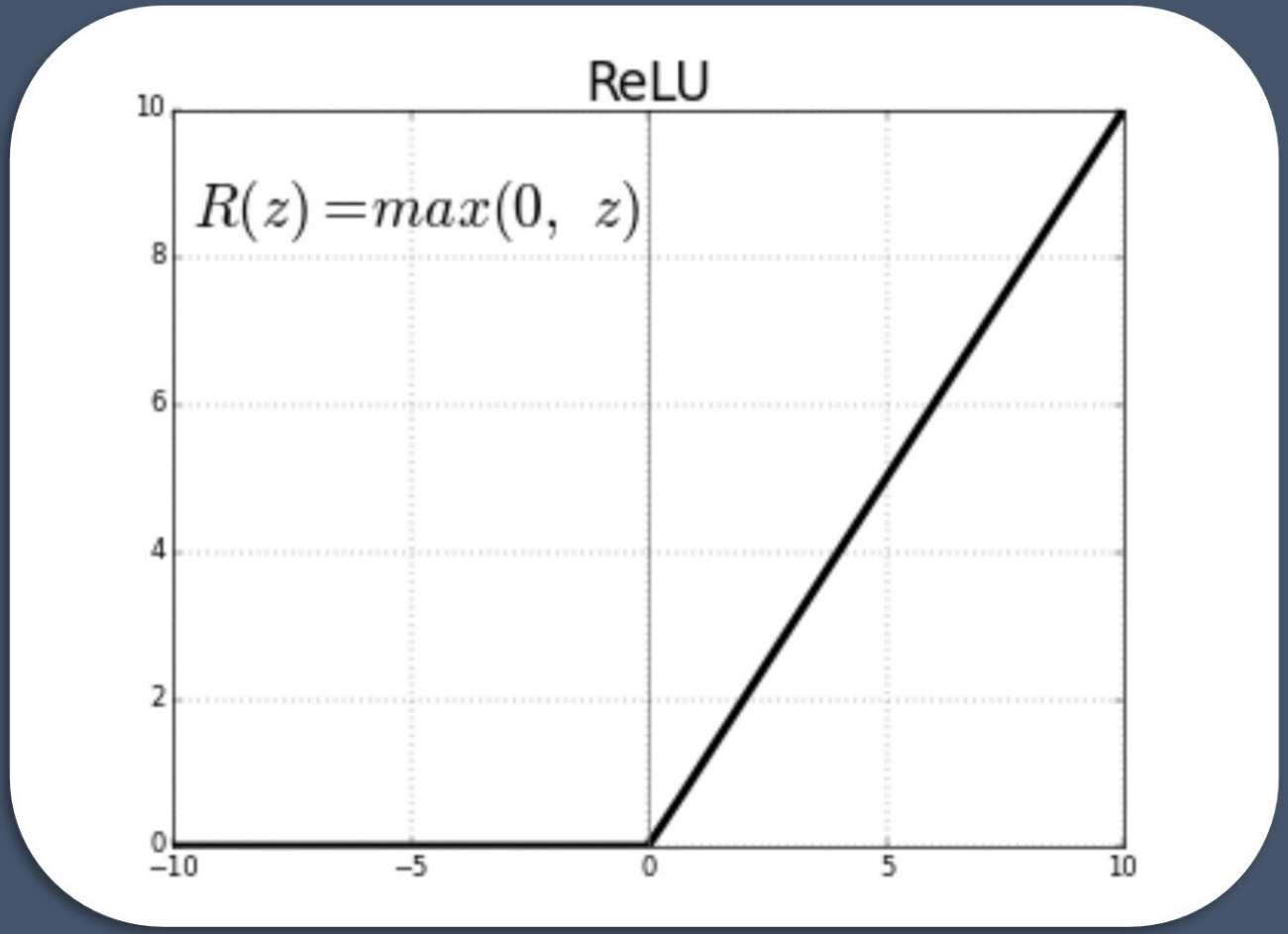
Why do inputs and outputs matter?




TensorFlow
Install
Develop
API r1.4
Deploy
Extend
Community
Versions
TFRC
get_collection
get_collection_ref
get_default_graph
get_default_session
get_local_variable
get_seed
v1 = foo() # Creates v.
v2 = foo() # Gets the same, existing v.
assert v1 == v2
If initializer is None (the default), the default initializer passed in the variable scope will be used. If that one is None too, a glorot_uniform_initializer will be used. The initializer can also be a Tensor, in which case the variable is initialized to this value and shape.


tf.optimizers
tf.profiler
tf.quantization
tf.queue
tf.ragged
tf.random
tf.raw_ops
tf.saved_model
tf.sets
tf.signal
tf.sparse
tf.strings
tf.summary
tf.sysconfig
tf.keras.initializers.GlorotNormal
See Stable
See Nightly
View source on GitHub
The Glorot normal initializer, also called Xavier normal initializer.
Inherits From: VarianceScaling, Initializer
View aliases
tf.keras.initializers.GlorotNormal(
Methods
from_config
get_config
__call__

Kaiming (He) Initialization
for activation functions like Relu



Kaiming He

FOLLOW

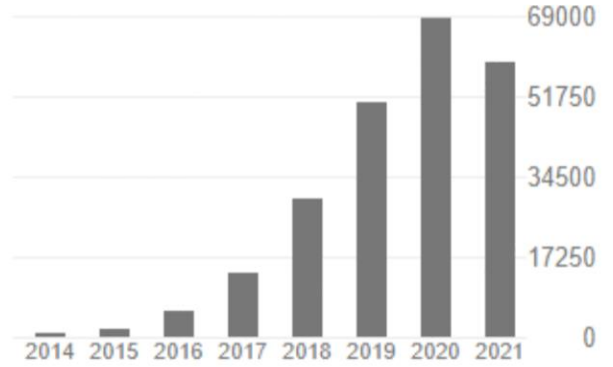
Research Scientist, Facebook AI Research (FAIR)
Verified email at fb.com - Homepage

Computer Vision Machine Learning

TITLE	CITED BY	YEAR
Deep Residual Learning for Image Recognition K He, X Zhang, S Ren, J Sun Computer Vision and Pattern Recognition (CVPR), 2016	90286	2016
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks S Ren, K He, R Girshick, J Sun Neural Information Processing Systems (NIPS), 2015	33938	2015
Mask R-CNN K He, G Gkioxari, P Dollár, R Girshick International Conference on Computer Vision (ICCV), 2017	14247	2017
Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification K He, X Zhang, S Ren, J Sun International Conference on Computer Vision (ICCV), 2015	13140	2015
Focal Loss for Dense Object Detection TY Lin, P Goyal, R Girshick, K He, P Dollár International Conference on Computer Vision (ICCV), 2017	9094	2017

Cited by

	All	Since 2016
Citations	234939	228368
h-index	58	57
i10-index	66	66



Public access VIEW ALL

0 articles 10 articles

not available available

Based on funding mandates

Weight Initialization in Neural Networks: A Journey From the Basics to Kaiming

James Dellinger Apr 3, 2019 · 11 min read

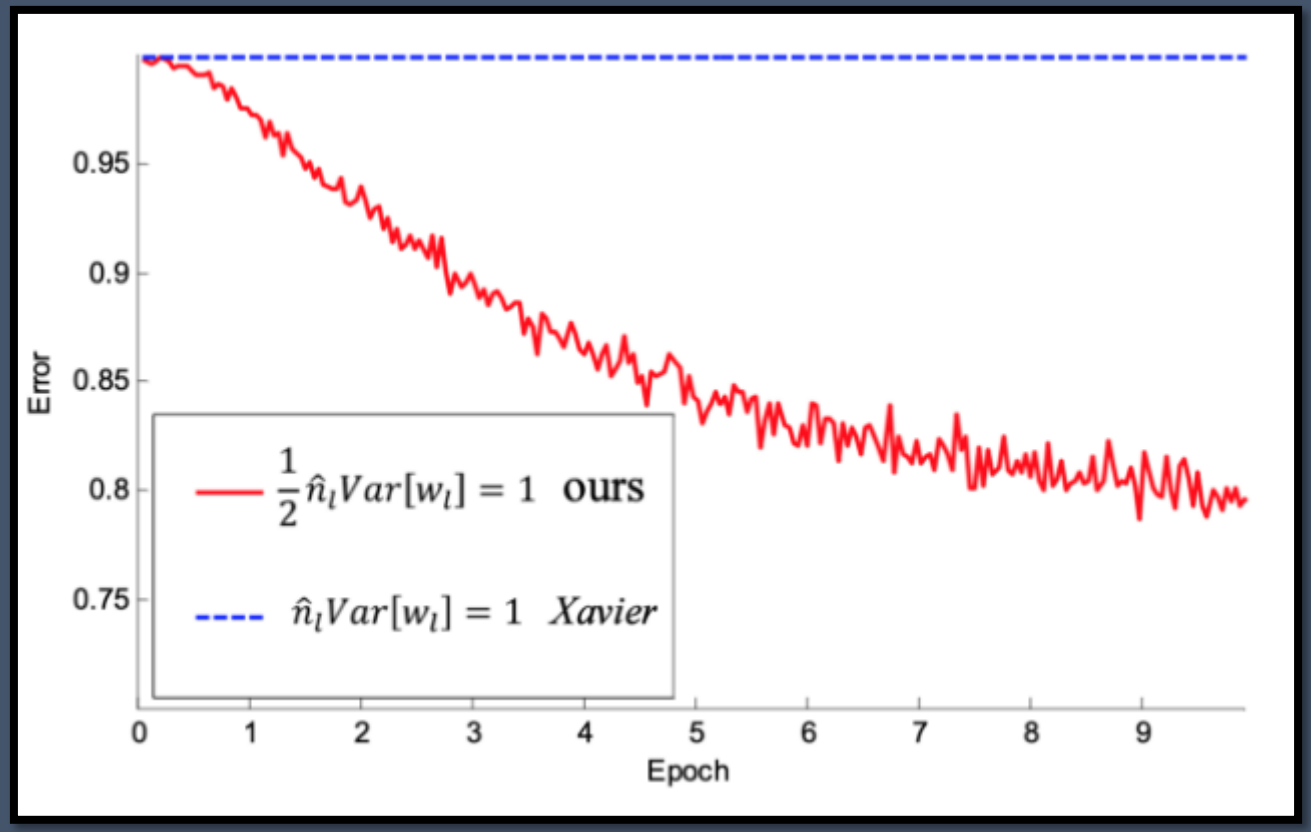
I'd like to invite you to join me on an exploration through different approaches to initializing layer weights in neural networks. Step-by-step, through various short experiments and thought exercises, we'll discover why adequate weight initialization is so important in training deep neural nets. Along the way we'll cover various approaches that researchers have

https://towardsdatascience.com/weight-initialization-in-neural-networks-a-journey-from-the-basics-to-kaiming-954fb9b47c79

In their 2015 paper, He et. al. demonstrated that deep networks (e.g. a 22-layer CNN) would converge much earlier if the following input weight initialization strategy is employed:

1. Create a tensor with the dimensions appropriate for a weight matrix at a given layer, and populate it with numbers randomly chosen from a standard normal distribution.
2. Multiply each randomly chosen number by $\sqrt{2}/\sqrt{n}$ where n is the number of incoming connections coming into a given layer from the previous layer's output (also known as the "fan-in").
3. Bias tensors are initialized to zero.

Incidentally, when they trained even deeper networks that used ReLUs, He et. al. found that a 30-layer CNN using Xavier initialization stalled completely and didn't learn at all. However, when the same network was initialized according to the three-step procedure outlined above, it enjoyed substantially greater convergence.



Stay Tuned
Data Science course has a lot of fun for devoted students