Medical Imaging & Intelligent Reality Lab.
Convergence Medicine/Radiology

The Forward-Forward Algorithm: Some Preliminary Investigations

Git: None

Presenter: Sunggu Kyung

Email: babbu3682@gmail.com

Paper Contents

1. Abstract
2. Introduction
3. Methods
4. Experiments
5. Conclusion

Background
Back-propagation

Introduction

Backpropagation (오차역전파)

$$f(x, y, z) = (x + y)z$$

x=-2, y=5, z=-4

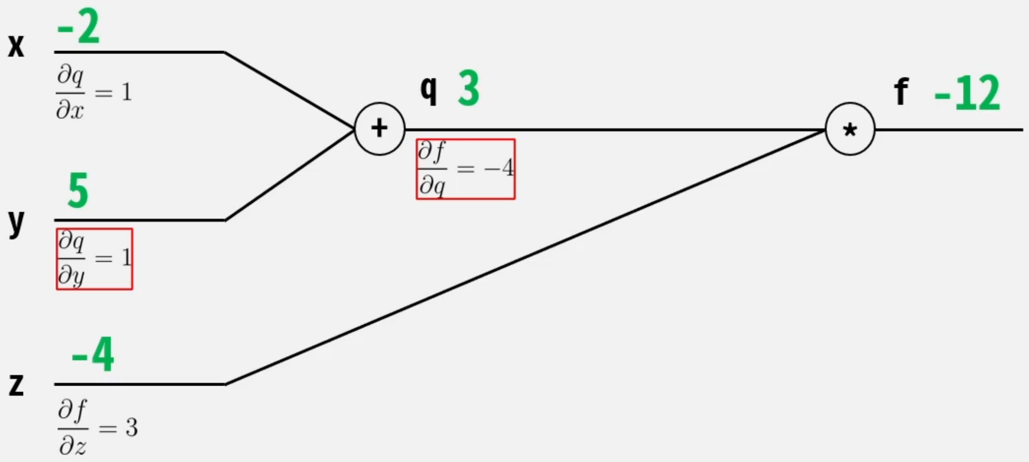
$$q = x + y$$

$$f = qz$$

Chain Rule

$$\frac{\partial f}{\partial x} = -4$$

$$\boxed{\frac{\partial f}{\partial y} = \frac{\partial f}{\cancel{\partial q}} \frac{\cancel{\partial q}}{\partial y} = (-4)(1) = -4}$$



The Forward-Forward Algorithm: Some Preliminary Investigations

Geoffrey Hinton
Google Brain
geoffhinton@google.com

Abstract

- The aim of this paper is to introduce a **new learning procedure** for neural networks and to demonstrate that it works well enough on a few small problems to be worth further investigation.

- The Forward-Forward algorithm replaces the forward and backward passes of backpropagation by two forward passes, **one with positive (i.e. real) data and the other with negative data** which could be generated by the network itself.

- **Each layer has its own objective function** which is simply to have high goodness for positive data and low goodness for negative data.

- The sum of the squared activities in a layer can be used as **the goodness** but there are many other possibilities, including minus the sum of the squared activities.

- If the positive and negative passes could be separated in time, **the negative passes could be done offline**, which would make the learning much simpler in the positive pass and allow video to be pipelined through the network **without ever storing activities or stopping to propagate derivatives**.

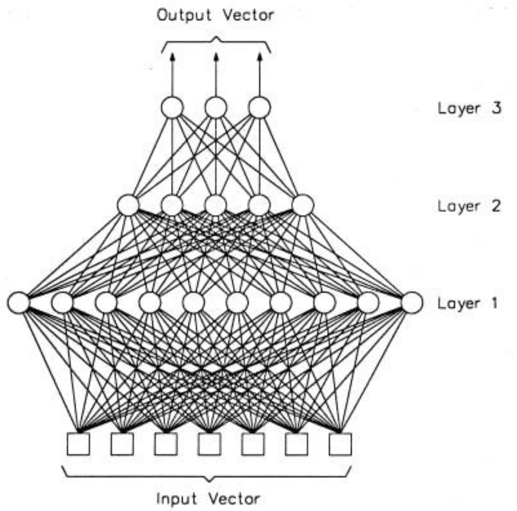
1. What is wrong with backpropagation

- As a model of how cortex learns, **backpropagation remains implausible.**

- Backpropagation through time as a way of **learning sequences is especially implausible**.

- A further serious limitation of backpropagation is that **it requires perfect knowledge of the computation** performed in the forward pass

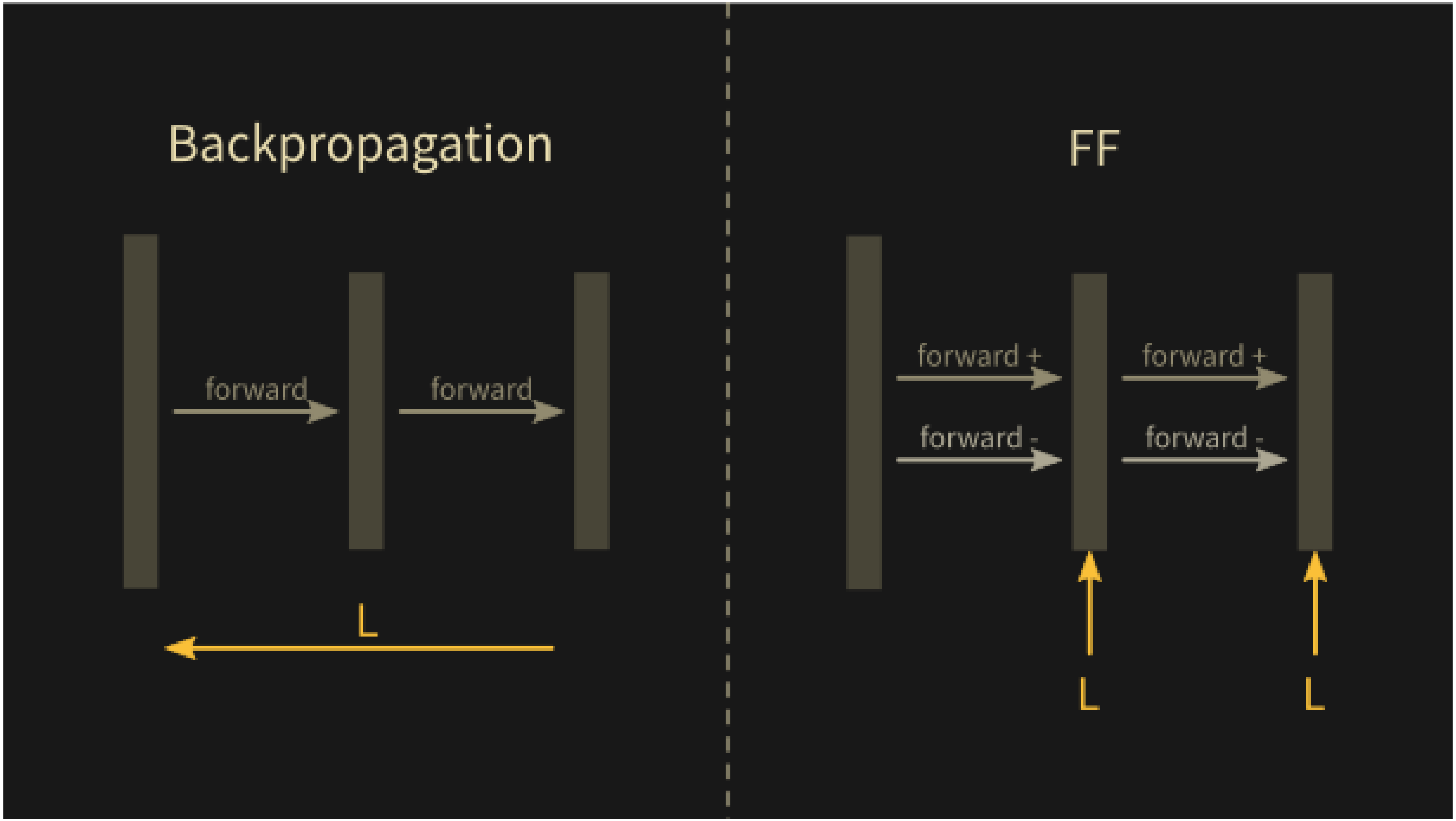
- In the absence of a perfect model of the forward pass, it is always possible to resort to one of the many forms of reinforcement learning.



1. What is wrong with backpropagation

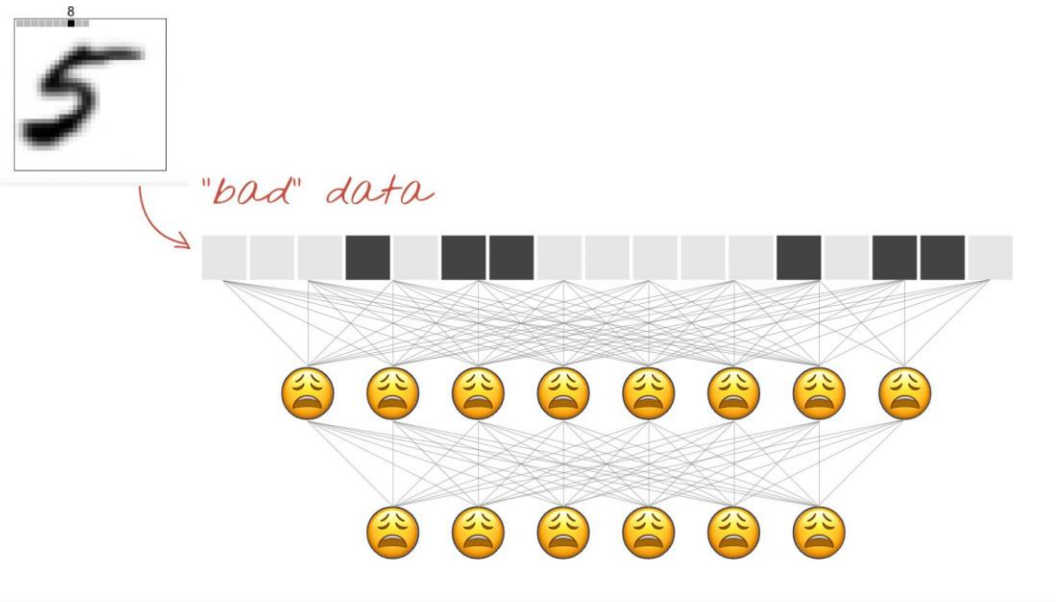
- The Forward-Forward algorithm (FF) has the advantage that it can be used when the precise details of **the forward computation are unknown**
- it can learn while pipelining sequential data through a neural network **without ever storing the neural activities or stopping to propagate error derivatives.**
- The forward-forward algorithm is somewhat slower than backpropagation and does does not generalize quite as well on several of the toy problems investigated in this paper so **it is unlikely to replace backpropagation** for applications where power is not an issue.
- The two areas in which the forward-forward algorithm may be superior to backpropagation are as a model of **learning in cortex** and as a way of making use of very **low-power analog hardware** without resorting to reinforcement learning.

2. The Forward-Forward Algorithm



2. The Forward-Forward Algorithm

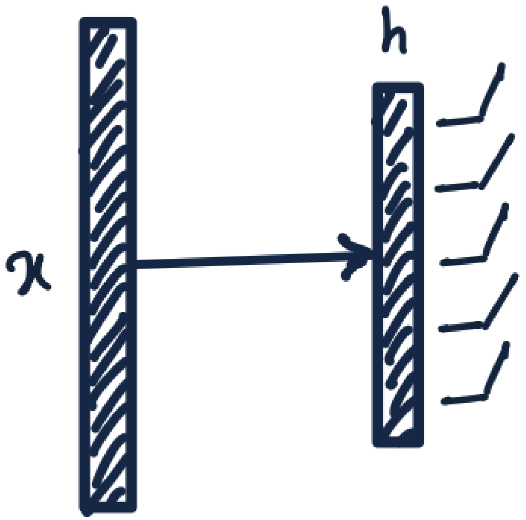
Good Case



Bad Case

2. The Forward-Forward Algorithm

$$p(positive) = \sigma\left(\sum_j y_j^2 - \theta\right)$$



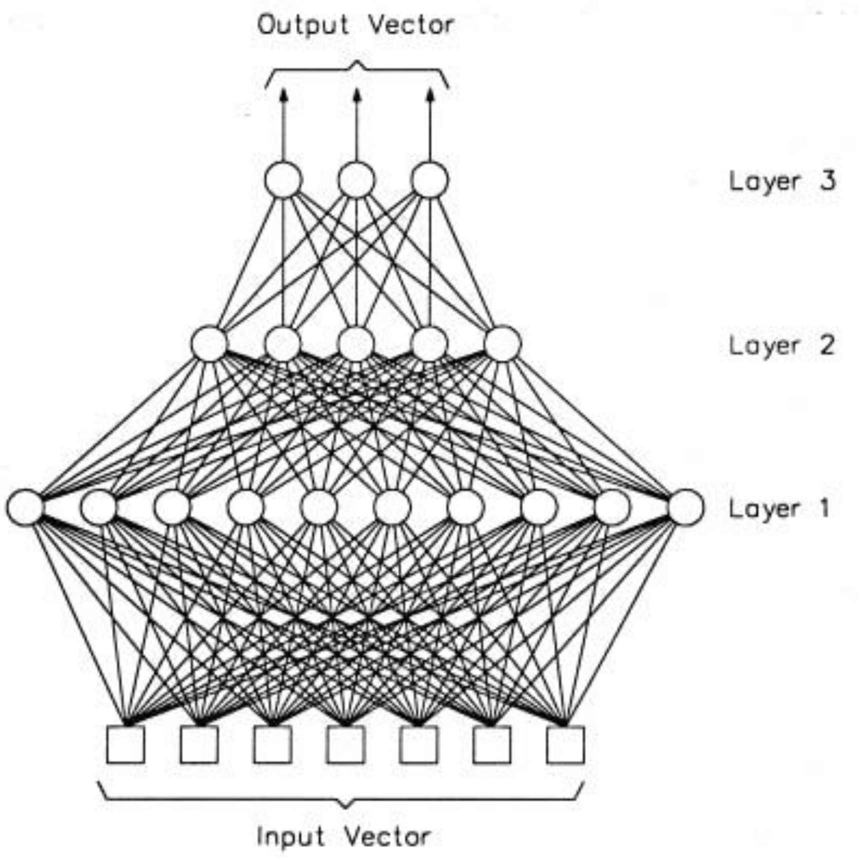
Goodness

$$G(x) := \sum relu(h_i)^2$$

We want:

$$G(x_{pos}) \gg threshold$$

$$G(x_{neg}) \ll threshold$$



2. The Forward-Forward Algorithm

- But if the activities of the first hidden layer are then used as input to the second hidden layer, it is trivial to distinguish positive from negative data by simply using the length of activity vector in the first hidden layer.

- There is no need to learn any new features. To prevent this, FF normalizes the length of the hidden vector before using it as input to the next layer.

- This removes all of the information that was used to determine the goodness in the first hidden layer and forces the next hidden layer to use information in the relative activities of the neurons in the first hidden layer. These relative activities are unaffected by the layer-normalization.

- To put it another way, the activity vector in the first hidden layer has a length and an orientation. The length is used to define the goodness for that layer and only the orientation is passed to the next layer.

2. The Forward-Forward Algorithm

- The Forward-Forward algorithm is a greedy multi-layer learning procedure inspired by **Boltzmann machines** and **Noise Contrastive Estimation**.

- The idea is to **replace the forward and backward passes of backpropagation by two forward passes** that operate in exactly the same way as each other, but on different data and with opposite objectives.

- The positive pass operates on real data and adjusts the weights to increase the goodness in every hidden layer. The negative pass operates on "negative data" and adjusts the weights to decrease the goodness in every hidden layer.

- The aim of the learning is to make the goodness be well above some threshold value for real data and well below that value for negative data.

- There are two main reasons for using the squared length of the activity vector as the goodness function. First, it has very simple derivatives. Second, layer normalization removes all trace of the goodness

3. Some experiments with FF

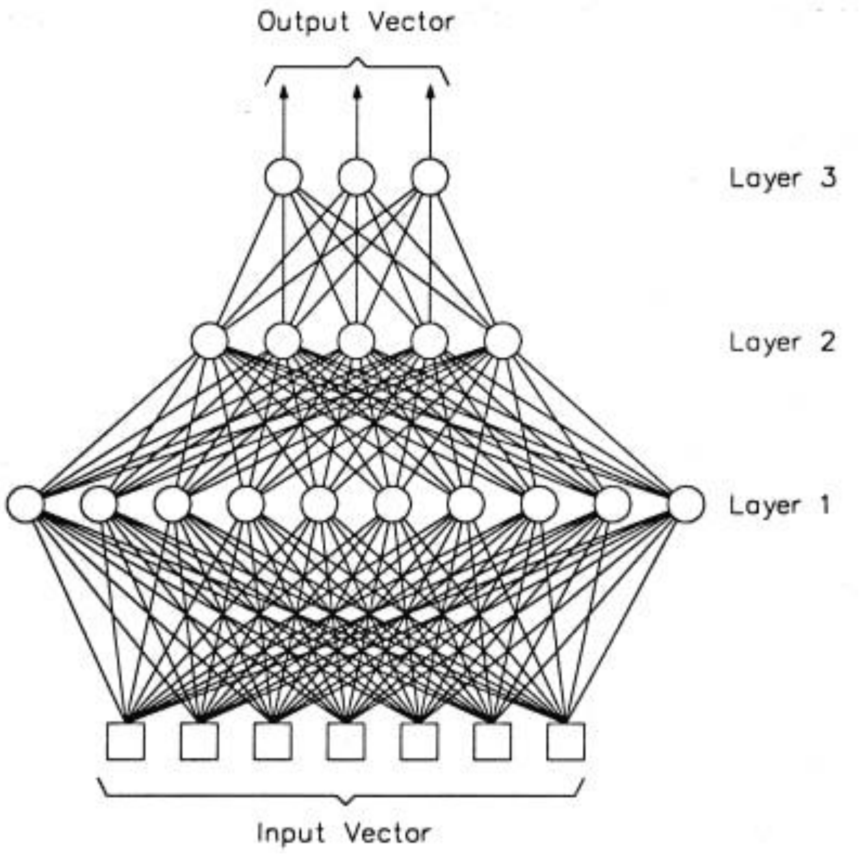
- MNIST

- First, if we have a good source of negative data, does it learn effective multi-layer representations that capture the structure in the data?

- Second, where does the negative data come from?

3. Some experiments with FF

- **A simple unsupervised example of FF**



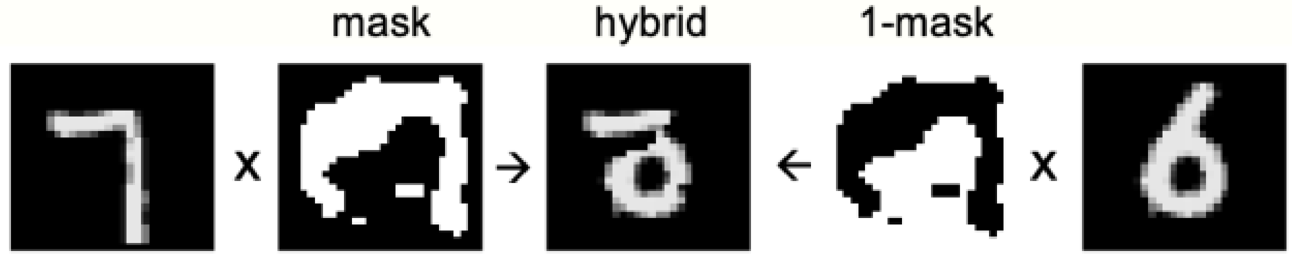


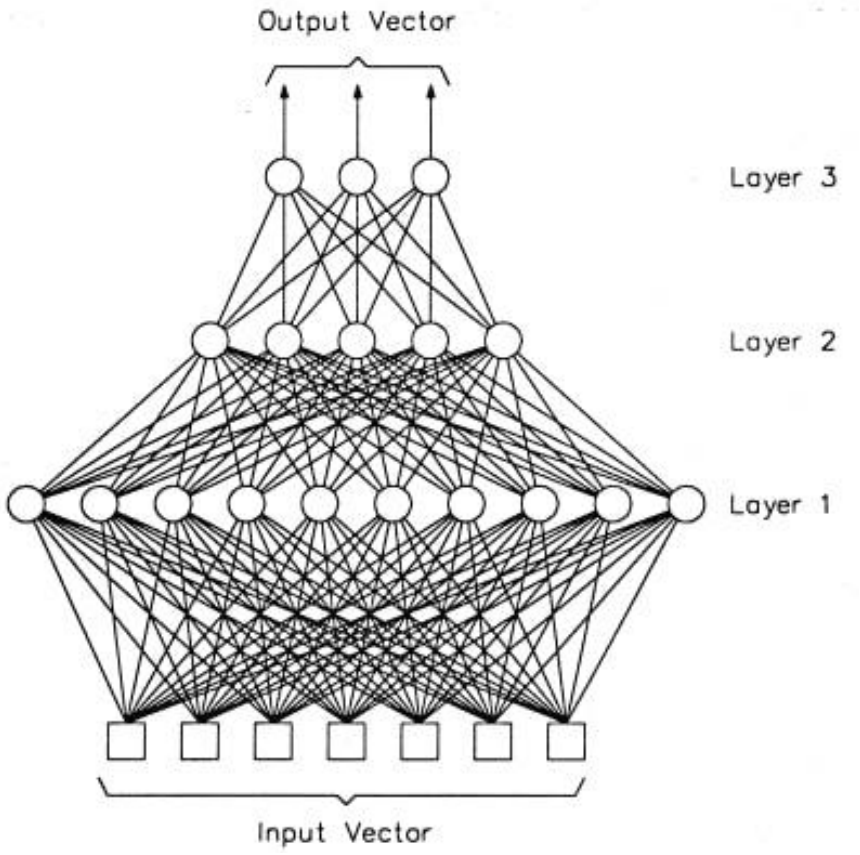
Figure 1: A hybrid image used as negative data

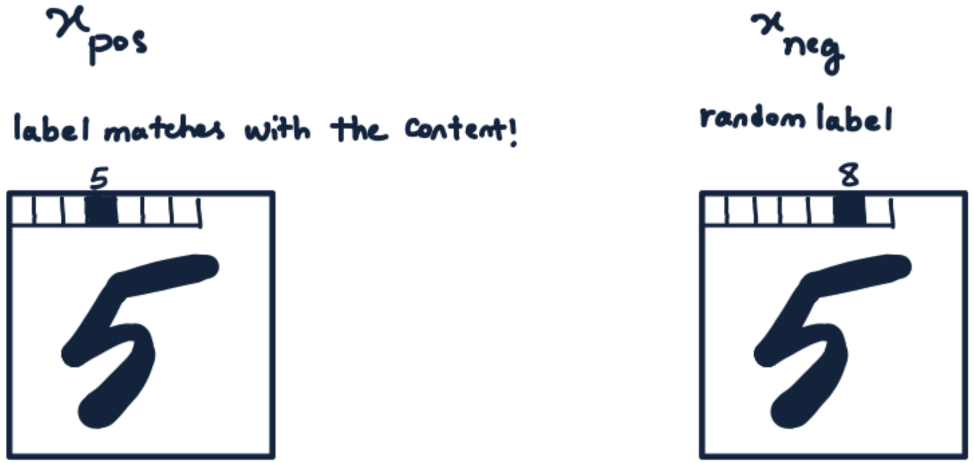
- After training a network with four hidden layers of 2000 ReLUs each for **100 epochs**, we get a **test error rate of 1.37%** if we use the normalised activity vectors of the last three hidden layers as the inputs to a softmax that is trained to predict the label.

- Using the first hidden layer as part of the input to the linear classifier makes the test performance worse.

- Instead of using fully connected layers we can use local receptive fields (without weight-sharing) and this improves the performance. After training for 60 epochs it gave 1.16% test error.

- **A simple supervised example of FF**





A network with 4 hidden layers each containing 2000 ReLUs and full connectivity between layers gets 1.36% test errors on MNIST after 60 epochs.

Backpropagation takes about 20 epochs to get similar test performance.

Doubling the learning rate of FF and training for 40 epochs instead of 60 gives a slightly worse test error of 1.46% instead of 1.36%.

3. Some experiments with FF

- **A simple supervised example of FF**

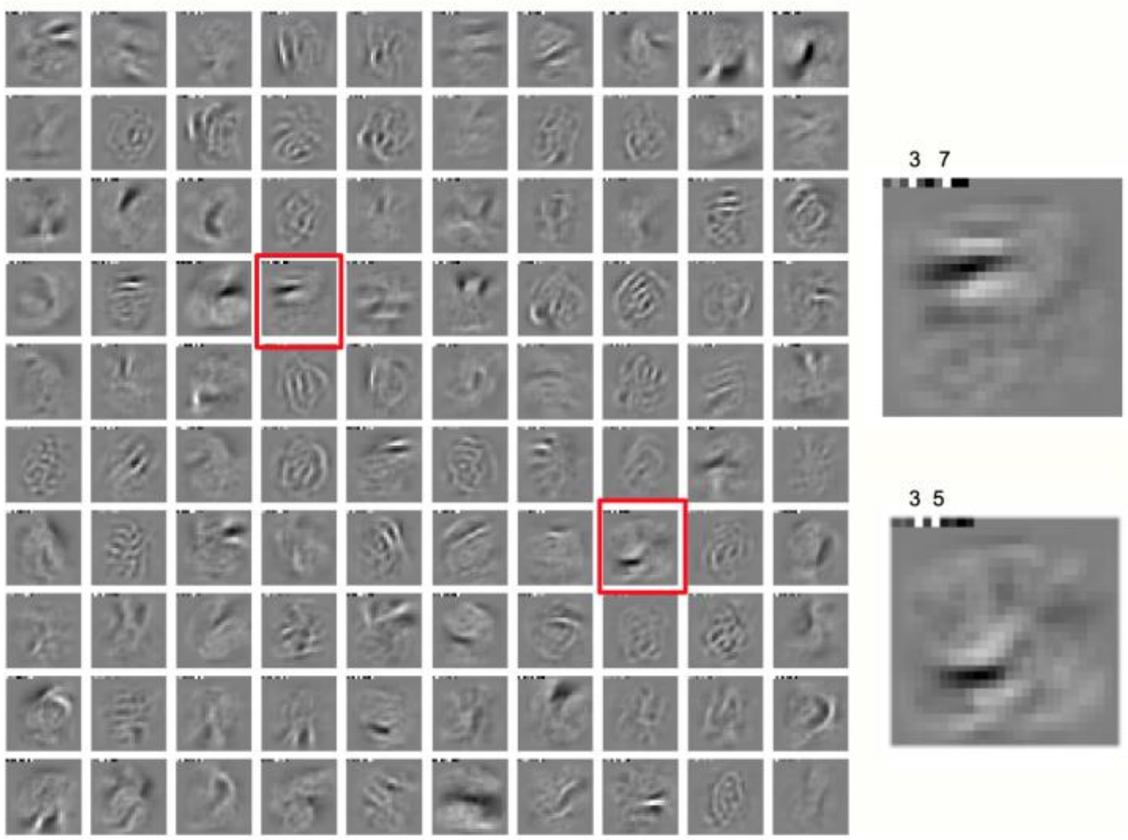


Figure 2: The receptive fields of 100 neurons in the first hidden layer of the network trained on jittered MNIST. The class labels are represented in the first 10 pixels of each image.

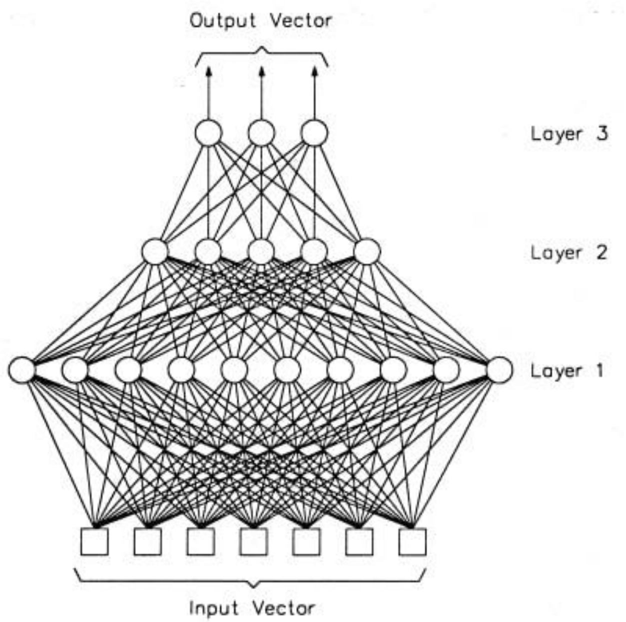
The Forward-Forward Algorithm

- **3.4 Using FF to model top-down effects in perception**

- What is learned in later layers cannot affect what is learned in earlier layers. This seems like a major weakness compared with backpropagation.

- The key to overcoming this apparent limitation of FF is to treat a **static image** as a rather boring video that is processed by a multi-layer recurrent neural network.

- FF runs forwards in time for both the positive and negative data, but, as figure 3 shows, the activity vector at each layer is determined by the normalized activity vectors at both the layer above and the layer below at the previous time-step.



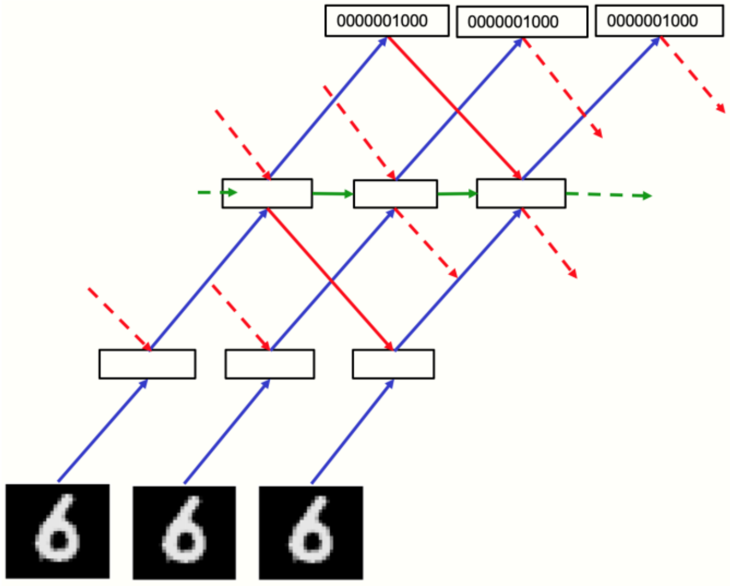
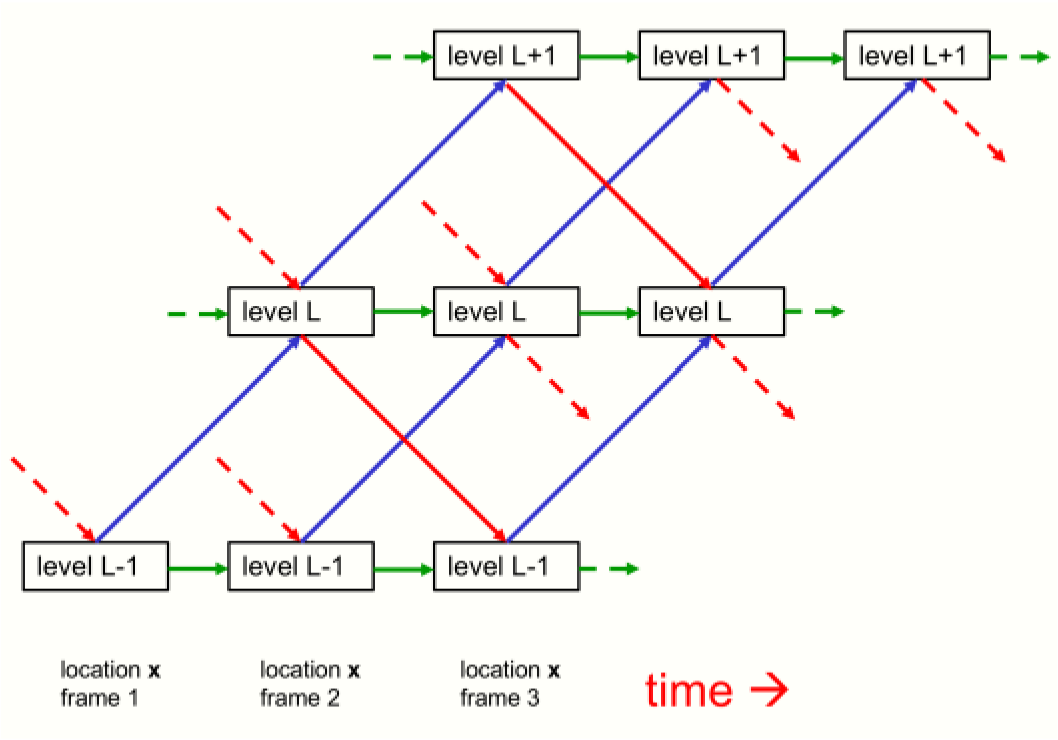


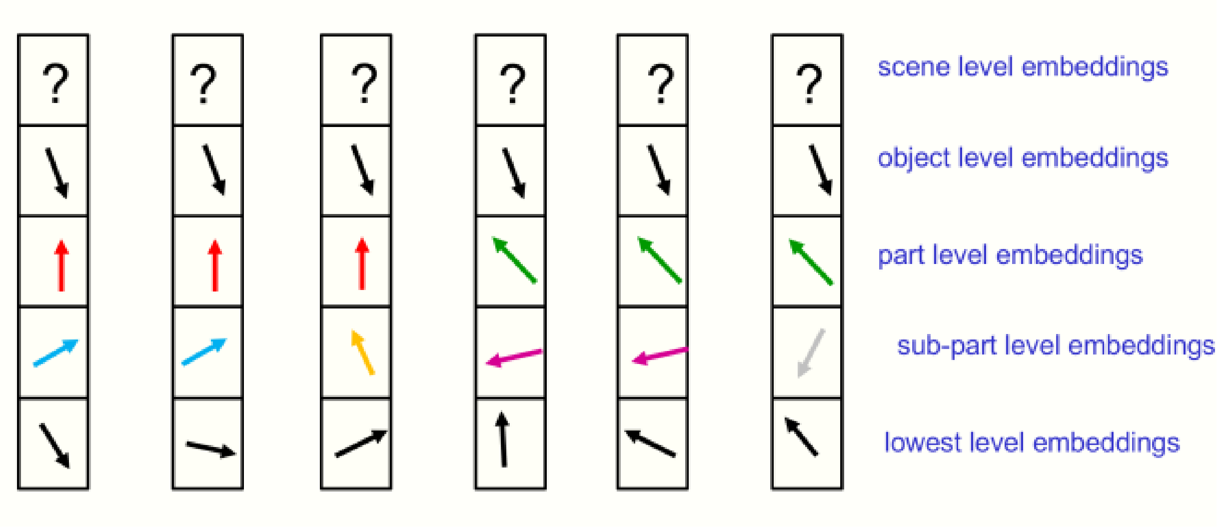
Figure 3: The recurrent network used to process video.

The Forward-Forward Algorithm

- **3.4 Using FF to model top-down effects in perception**

[2021] How to represent part-whole hierarchies in a neural network, Geoffrey Hinton



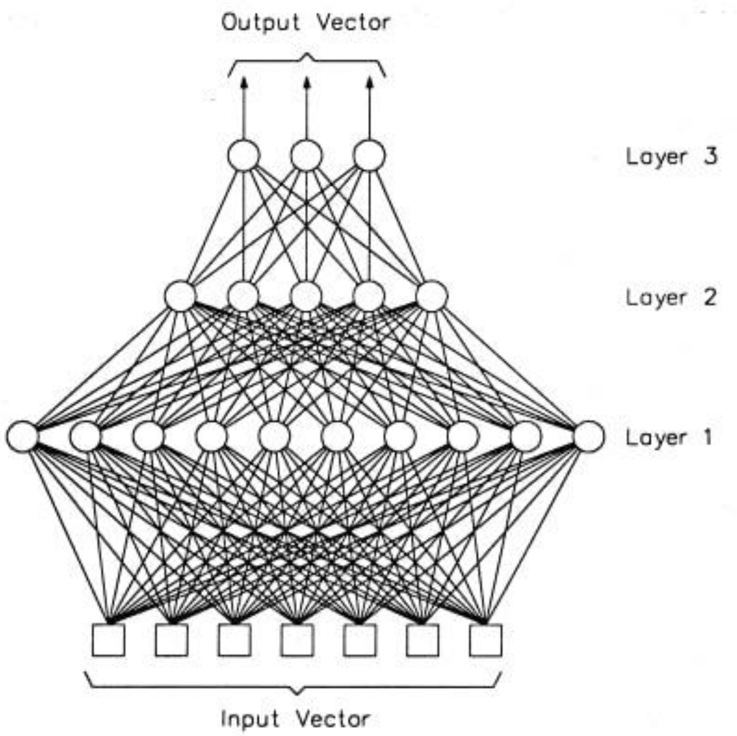


The Forward-Forward Algorithm

- **3.5 Using predictions from the spatial context as a teacher**

- In the recurrent net, the objective is to have **good agreement** between the input from the layer above and the input from the layer below for positive data and bad agreement for negative data

- The top-down input will be determined by a larger region of the image and will be the result of more stages of processing so it can be viewed as a contextual prediction for what should be produced by the bottom-up input which is based on a more local region of the image.



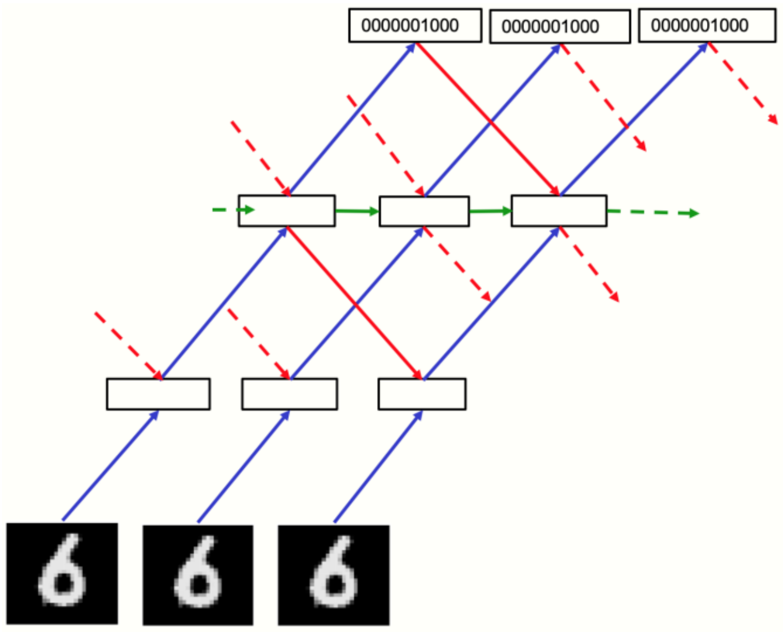


Figure 3: The recurrent network used to process video.

The Forward-Forward Algorithm

- **4. Experiments with CIFAR-10**

learning procedure	testing procedure	number of hidden layers	training % error rate	test % error rate
BP		2	0	37
FF min ssq	compute goodness for every label	2	20	41
FF min ssq	one-pass softmax	2	31	45
FF max ssq	compute goodness for every label	2	25	44
FF max ssq	one-pass softmax	2	33	46
BP		3	2	39
FF min ssq	compute goodness for every label	3	24	41
FF min ssq	one-pass softmax	3	32	44
FF max ssq	compute goodness for every label	3	21	44
FF max ssq	one-pass softmax	3	31	46

The Forward-Forward Algorithm

- **4.1 A simple sequence learning example of FF**

- Consider the task of learning to predict the next character in a string from the previous 10 characters.

- One simple way to do this is to use several hidden layers to extract higher-order features of the preceding 10 character string and then to use the activities of the hidden units in these layers as inputs to a softmax that predicts the probability distribution over all possible next characters.

- The hidden layers can be trained by using strings of 10 characters from the real data as the positive data and strings in which the last character has been replaced by a prediction from the previous 10 characters as the negative data.

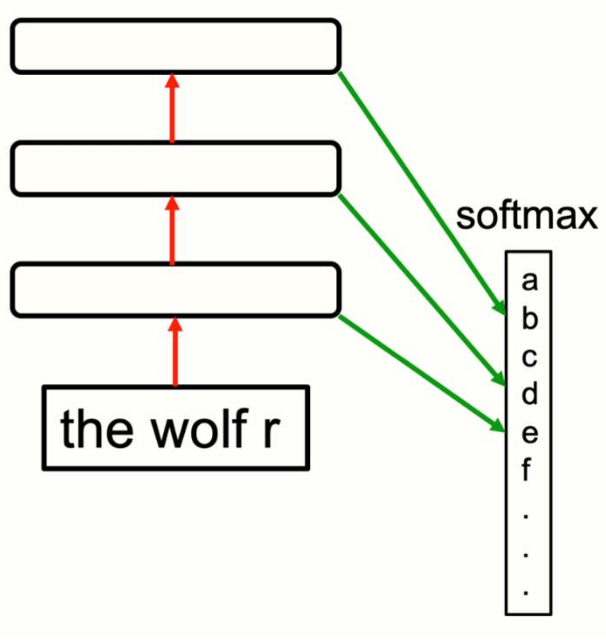
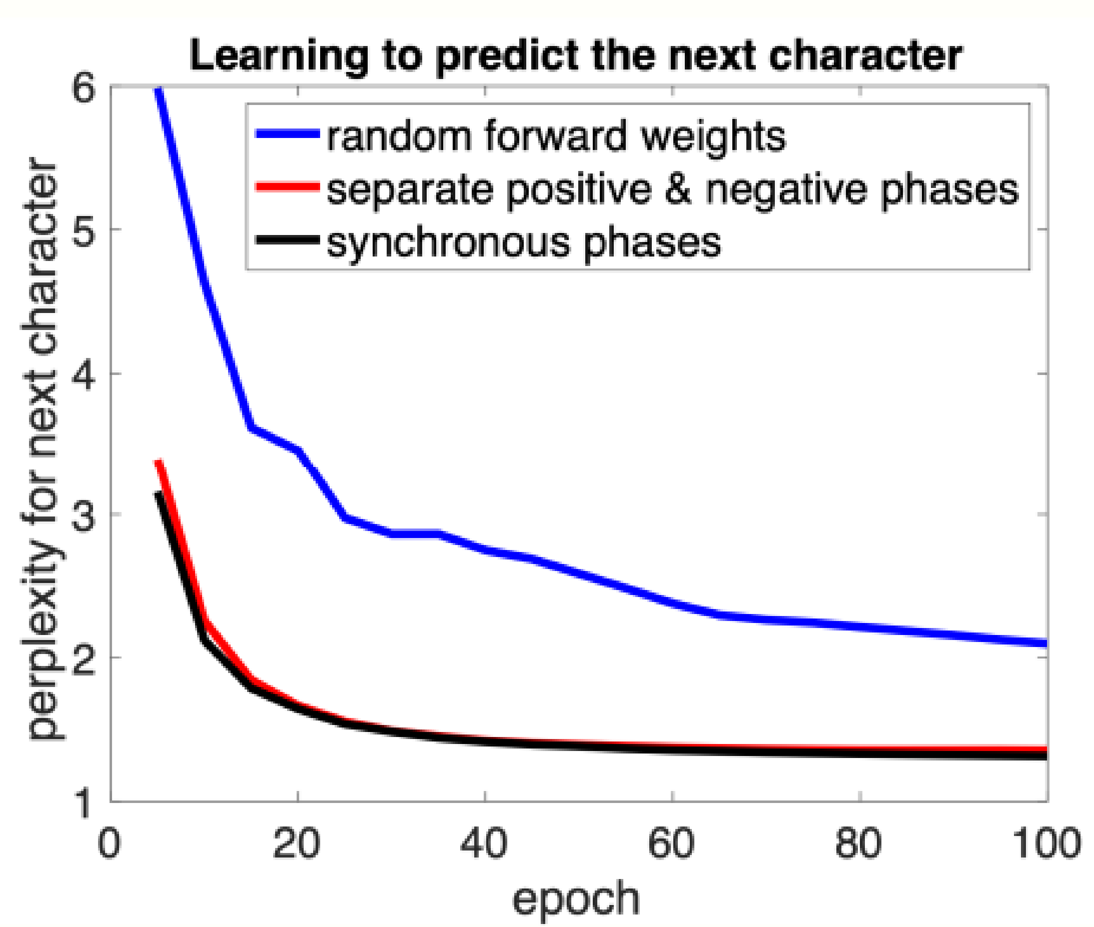


Figure 4: The net used to predict the next character.

The Forward-Forward Algorithm

- **4.1 A simple sequence learning example of FF**



Reference

- https://www.youtube.com/watch?v=rVzDRfO2sgs

- Code:
- https://github.com/ghadialhajj/FF_unsupervised/blob/master/main.py#L142
- https://github.com/mohammadpz/pytorch_forward_forward/blob/main/main.py

Thank you for your Attention….!