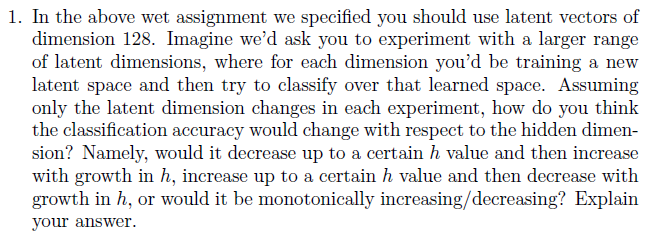
{ Final Project – Dry }

Submitted by :

|  |  |  |
| --- | --- | --- |
| Name | ID | Email |
| Husain Azaqy | 213382104 | Hssain.azaqy@campus.technion.ac.il |
| Tomer Danilin | 207933961 | tomerdanilin@campus.technion.ac.il |

Note: For checking convenience , the questions where copied to this file (Black frames)



**Answer:**

We expect the classification accuracy to peak at a certain h value, from there the accuracy would either plateau (no improvements will be made) or even drop for way too large h since classifying over a larger latent space should be harder beyond some point ,due to the classifier’s compute being fixed in this context.

Since the latent space is a learned alternative representation of the signal domain, given a fixed encoder architecture, we expect that up to some h the learned latent space will get richer, beyond that h we think the learned representation can have redundancies and even get too complex to classify over.

A close up of a text

AI-generated content may be incorrect.

**Answer:**

1. UAT: for any loss criterion that is reasonable **(\*\*\*)** , there exists a finite width fully connected neural network of depth 1 with activation that is non -polynomial, bounded and continuous such that:

Is Optimal.

**(\*\*\*):** Theloss is continuous ,Proper (Consistent) meaning as the network output gets closer to the true function, the loss should approach zero. The loss is Differentiable (or Subdifferentiable) and is Convex.

These properties ensure that training a neural network is stable, efficient, and leads to meaningful learning. Yet some assumptions can be ignored in practice.

This theorem directly supports the claim of the friend in question. Note that this is the general format of the theorem, the original theorem is for uniform convergence (supremum of differences between outputs approaches zero).

1. The conclusion is wrong due to many practical reasons:
2. The theorem provides an exponential bound on the width of the single layer network, which is largely impractical. That is due to optimization and memory limits. This is known as the curse of dimensionality.
3. Not only it is possible to “compensate” for lack of network width by depth, the added activations between layers help find more complex relations more efficiently. And the structure of the relations learned by each layer are more hierarchical and explainable. Early layers tend to learn simple features while later layers tend to extract more complex and robust features and relations.

A white text on a white background

AI-generated content may be incorrect.

**Answer:**

1. The advantages of using a CNN over an MLP are as follows:

|  |  |  |
| --- | --- | --- |
| Criterion | CNN | MLP |
| Parameter count | Uses shared weights and local connections, reducing parameters. | Fully connected layers make them memory heavy. |
| Translation invariance or equivariance | Recognize objects regardless of position due to pooling and convolution. | Lack this property and are sensitive to position changes. |
| Spatial considerations | Automatically extract spatial features like edges and patterns. Have a receptive field that scales with depth. | Treat all pixels as independent, ignoring spatial relationships. |
| Computational Efficiency | Faster due to weight sharing in the form of filters/kernels and generally lower hidden dimensions. | Requires more memory and computation. |
| Scalability to Large Images | Efficiently process high-dimensional images. | Become impractical as image size increases. |

1. We don’t agree with Alice , like mentioned above, yes the convolution operation is a linear operation, but it is applied differently, for each filter, the weights are universal no matter where it is applied on the image. This is because it is applied with a sliding window fashion which allows it to capture more spatial features.

Compared to an MLP, which seemingly treats all pixels as independent ignoring any spatial relationships.

A close up of text

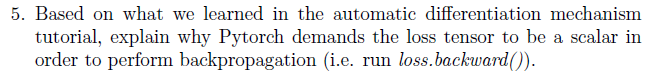
AI-generated content may be incorrect.

**Answer:** We compare both approaches as follows:

|  |  |  |
| --- | --- | --- |
| Aspects | Exponential Moving Average | Linear Averaging |
| Responsiveness to recent updates in gradients | More responsive to recent gradients due to exponential decay of past gradients. | Less responsive, as the decay is linear and gradual. |
| Memory limitations | Requires storing only the current and previous values. | Requires storing all past gradients within the window. |
| Convergence speed and efficiency | Faster convergence due to better gradient adaptation. | Slower convergence, not as efficient due to slower adaptation. |
| Adaptation of learning rate | Allows for dynamic adjustment of learning rates (e.g., in Adam). | Less effective at adapting learning rates. |

To sum things up:

An exponential moving average is generally better for efficiency, convergence, and more responsive, while linear moving average tends to be less adaptive to recent gradients and slower for convergence.



**Answer:**

The loss tensor should be a scaler due to the recursive nature of the reverse mode of AD.Since in the reverse mode of AD we compute the gradients relevant to each node in the computational graph in reverse topological order. We multiply the previous derivative with the current one and update the current gradient. We need the multiplication to always compile therefore the loss is a scaler. If the loss was a matrix or a vector the derivative would be a matrix making the multiplications in this algorithm not possible.

A close up of text

AI-generated content may be incorrect.

**Answer:**

1. Inductive bias refers to the assumptions a model makes about the data in order to generalize better.
2. CNN’s mainly assume two things, translation invariance/equivariance and spatial locality. (In the context of image processing for example)

Translation invariance or equivariance refer to the idea that a translation of the input image should result in the same output in case of invariance and a so called “expected” change in the output in the case of equivariance.

Spatial locality is the underlying assumption that in order to process a given input pixel we should consider nearby pixels.

1. Generally speaking, training a model with inductive bias can be made more efficient and effective taking said bias into consideration. Yet sometimes the assumptions about the data don’t hold. Or even hold at training time but not at inference time. For example, say a computer vision model aimed to classify the existence of a cat in an image assumes all cats in all pictures appear in a similar scale. If said model leans into this bias it would perform very poorly when presented with pictures containing cats of different scale.

Inductive bias is good when the assumptions hold at training and inference, we want more of these biases. Yet they may hinder the performance of models given these assumptions don’t always hold.

A white text with black text

AI-generated content may be incorrect.

**Answer:**

1. The inner computation of is of complexity , it results in an matrix, dividing each entry by takes a complexity of . The rowwise SoftMax computation is for each row, meaning another for the entire matrix is . Finally since the matrix V is of shape , the final product would take .

In total the complexity is .

1. Since Q is of dimensions , Softmax of takes time, the multiplication is of complexity and results in a matrix. Finally the multiplication in the middle is of complexity

All in all, the complexity here is , given that , this is much preferred complexity wise compared to , effectively switching an n for a d multiplicatively. Yet, this doesn’t preserve the function of the attention formulation.

The original attention computation calculates similarity scores between all pairs of tokens by computing allowing for global interaction between tokens.

Then scales these scores and applies SoftMax, and the final multiplication weighs the values 𝑉 accordingly.

The new computation messes the global interaction between tokens by pre weighting the values V with the keys before taking the queries into consideration.

A close up of a map

AI-generated content may be incorrect.

A screenshot of a computer game

AI-generated content may be incorrect.

**Answer:**

1. Assuming the attention here is of a model translating some language to English, each pixel here is the attention weight assigned by the model to an English word for translating the given word in the other language to English.
2. The meaning of having a row with only one non zero pixel is that the model implies that the only relevant word in English for translating the word in the other language that corresponds to that row is the word with the non zero pixel in it’s column. For example, to translate 1992 the model implies it only needs to pay attention to the 1992 in English.

Essentially A row with only a single non zero pixel in the map means that the model is focusing entirely on one specific source word when generating the corresponding target word.

1. Having several non zero pixels in the same row means the model implies that the translation of the word in said row is dependant on multiple words in the sentence. Meaning the model is focusing on multiple sources to generate the target word.
2. If a row has only one non zero pixel then that pixel is white and has a weight of 1, since the sum of each row in the map is 1 due to the SoftMax, this means all other pixels are black and have an attention weight of 0. Since the scale of the map is in grey scale, if at least two pixels aren’t zero, then both all aren’t 1, therefore all pixels in that row are some shade of grey.

A white text with black text

AI-generated content may be incorrect.

**Answer:**

A text on a page

AI-generated content may be incorrect.

**Answer:**