Nic Kuo
u6424547@anu.edu.au

Discussing TCN: On Dilation and Padding

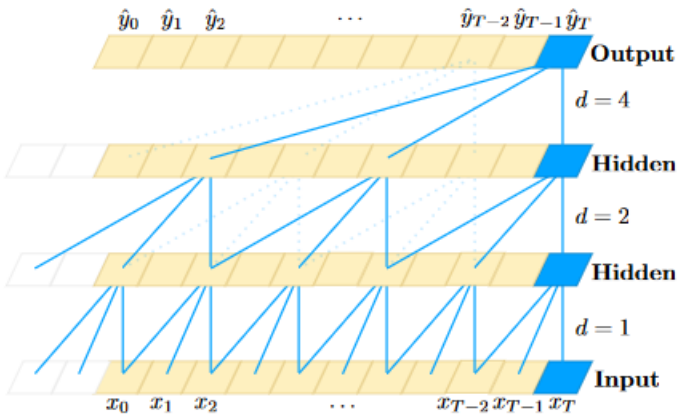
This document is meant to cover the rationale behind dilation and padding. In particular, we will discuss jerrybai1995's experimental setup for sequential MNIST.

In the last document (Doc001), we discussed the inner workings of **Causal_Conv1d**. We mentioned that the code

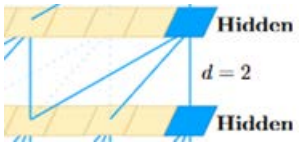
```
01.  class Causal_Conv1d (nn.Module):
02.      def __init__(self, n_inputs, n_outputs, kernel_size, padding):
03.          super(Causal_Conv1d, self).__init__()
04.          self.chomp_size = kernel_size - 1
05.          self.C = nn.Conv1d(n_inputs, n_outputs, kernel_size,
06.                             padding = self.chomp_size)
07.
08.      def forward(self, x):
09.          out  = self.C(x)
10.          out  = out[:, :, :-self.chomp_size].contiguous()
11.          return out
```

has 3 important features: padding, Conv1d, and chomping. However, we purposely left out dilation and we will focus on this aspect in this document.

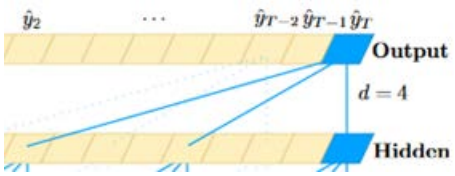
Dilation



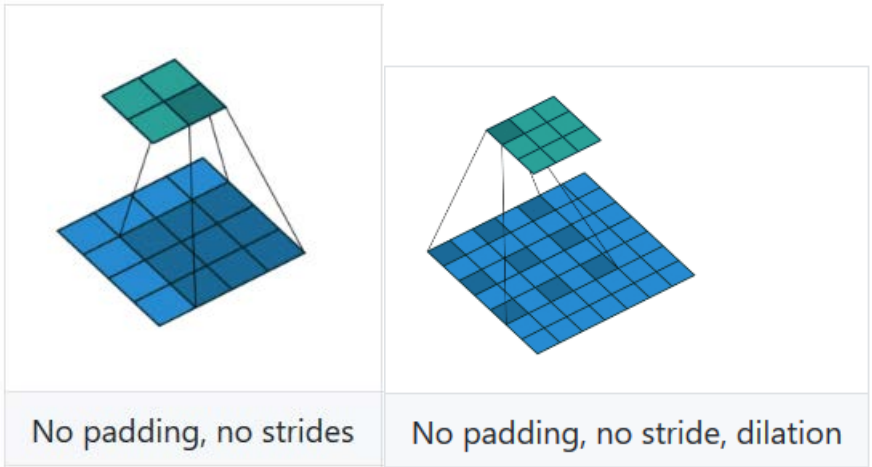
The above image can be found in Figure 1 (a) on page 4 of the original paper. As shown in the figure, not every data point is processed by the TCN.



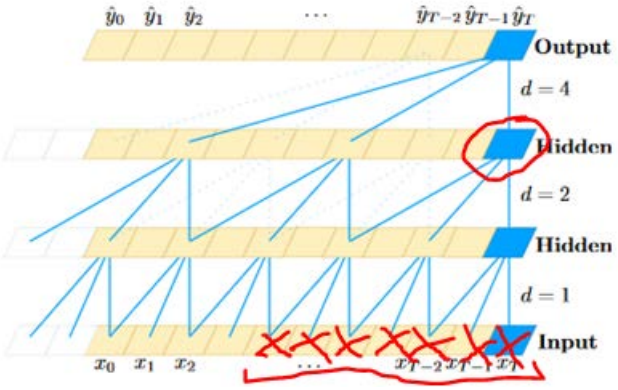
The second hidden layers has a gap between instances T



and $T - 2$; whereas the third layer has a gap between instances T and $T - 4$. These gaps are known as dilations, and they decide how sparsely the network samples data points from the input.



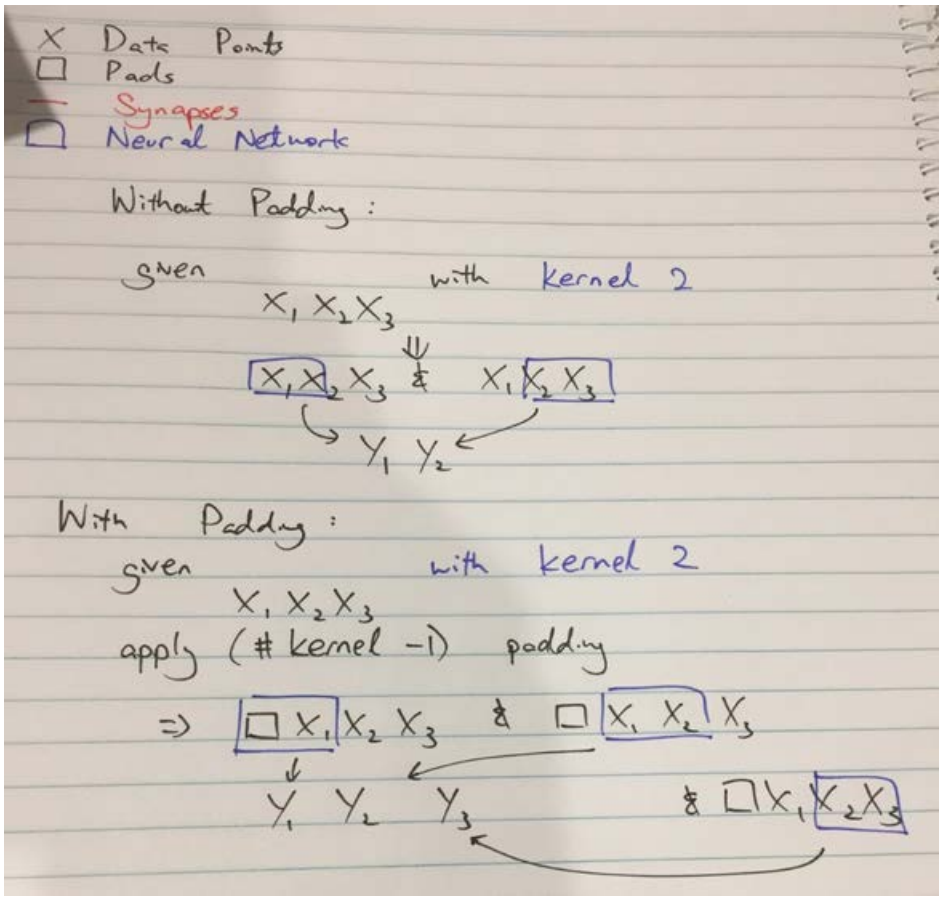
This concept can be demonstrated with more clarity in Conv2ds; and the above images can be found in the official Github page of Pytorch under https://github.com/vdumoulin/conv_arithmetic/blob/master/README.md.

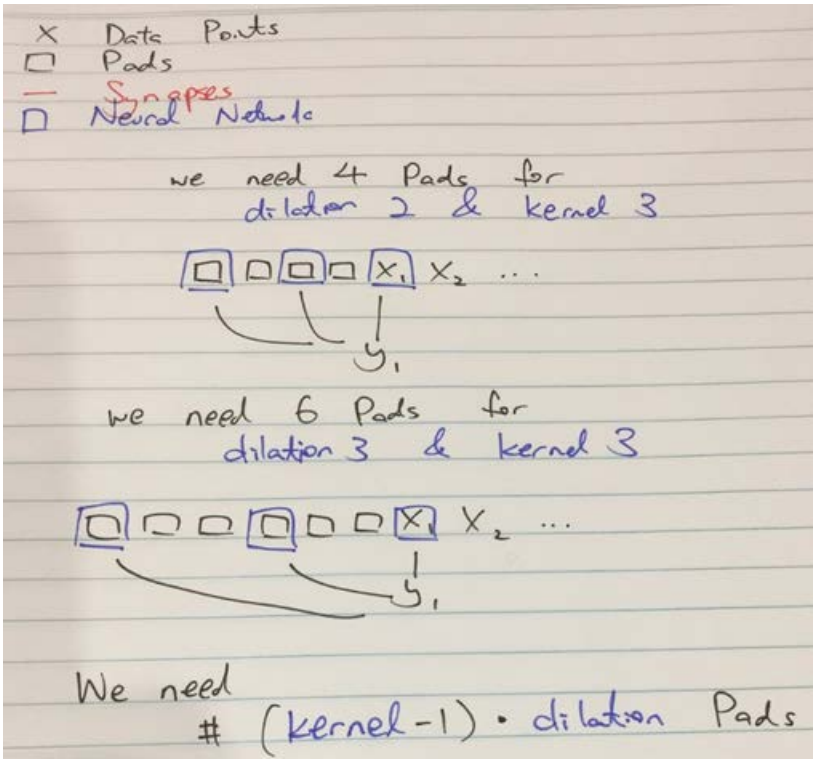


The purpose of using dilations is simple: to yield a design with a large receptive field while avoiding a complicated fully connected network. For instance, the neuron labelled with a red circle has access to latent presentations of all inputs labelled with a red cross despite not directly connecting with any of them.

Effects of Dilation on the Causal Convolutions

Dilation changes the amount of pads required for a **Causal_Conv1d**. In D001 we mentioned that padding is required to prevent future information from leaking into the current inferential process.





Now with dilation, things get a bit trickier and we need

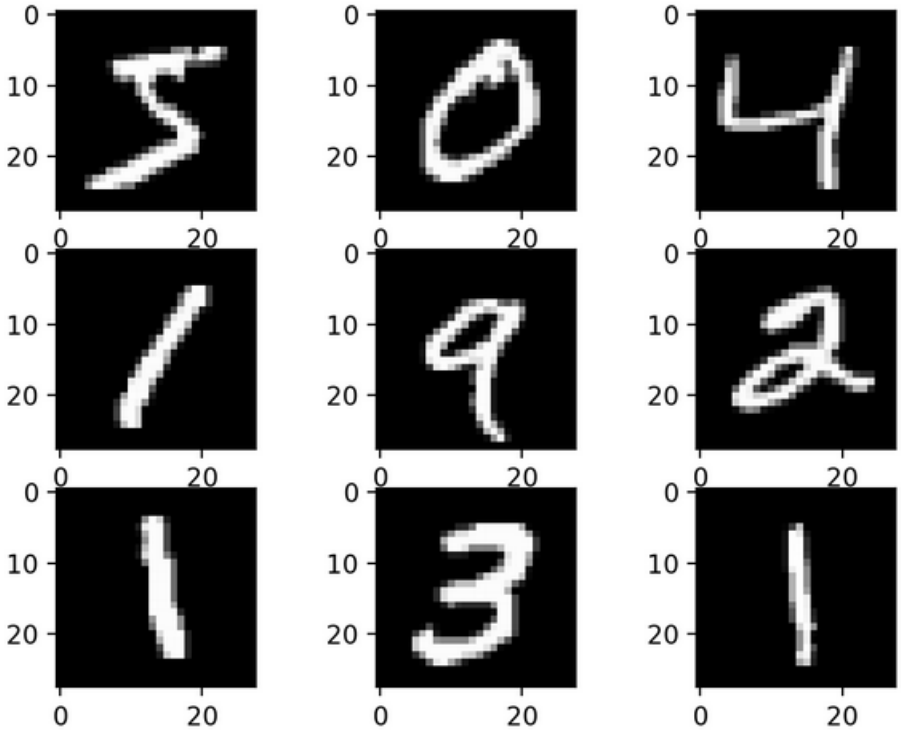
$$(\text{kernel} - 1)\,\text{dilation}$$

amount of pads. To emphasise, -1 because 1 receptive field of the kernel always take the 1st input data. Hence, we also need to replace the code with

```
04.         self.chomp_size = (kernel_size – 1) * dilation_size
05.         self.C = nn.Conv1d(n_inputs, n_outputs, kernel_size,
06.                            dilation  = dilation_size,
07.                            padding = self.chomp_size)
```

Starting next page, we will discuss the hyperparametric combinations used by jerrybai1995 for pixel MNIST.

The hyperparametric setup for pixel MNIST



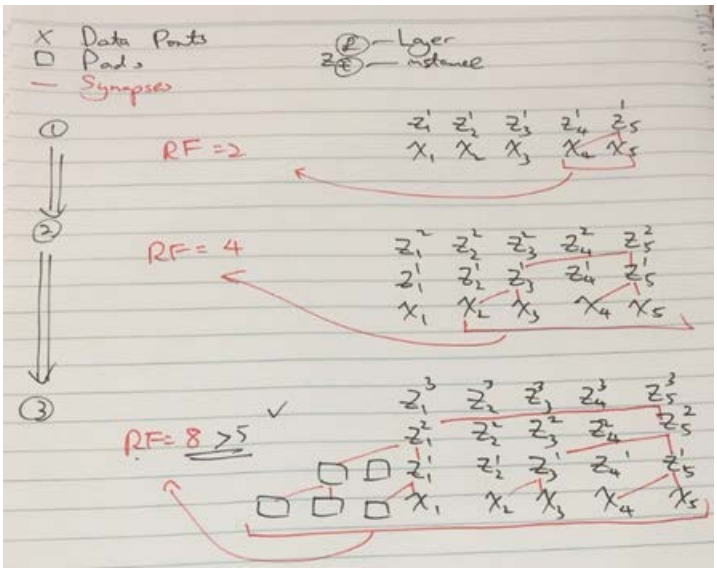
(source: Google)

MNIST is a famous computer vision dataset commonly used for benchmarking deep learning performances. As shown in the figure above, each image contains a handwritten digit of 0 – 9 and each are stored on a 28 x 28 image pixel box. The task that jerrybai1995 selected was pixel MNIST (pMNIST), where each image is presented column by column (or row by row) as a vector of 784(=28x28) pixels to the network for classification. This task was originally presented to test for the storage capacity for an RNN. That is, to test and see whether an RNN is capable of running across the lengthy 784 units of pixels and conduct classification.

In his codes in TCN / TCN / mnist_pixel / pmnist_test.py, jerrybai1995 employed 8 layers of **Causal_Conv1d** each with kernel_size 7 (note, dilation is defaulted as 2). We will discuss why this is required in the next page.

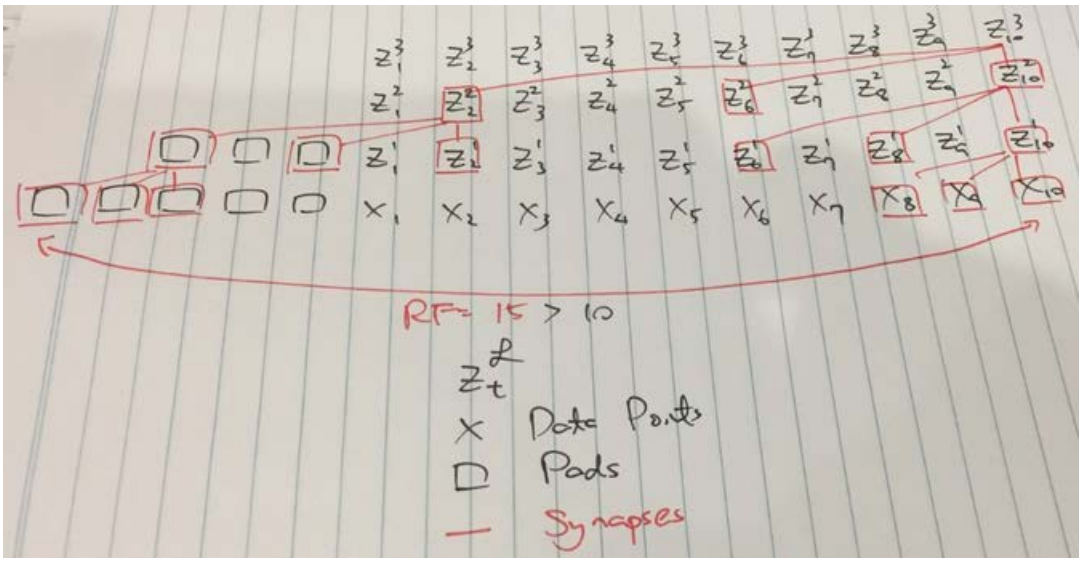
Let's start with a simpler question:

*How many **layers** do we need to yield a receptive field (RF) that fully covers a sequence of inputs* x_1, x_2, x_3, x_4, x_5
if this sequence were to be processed by a
***TCN with incremental dilation size 2 and kernel size 2**?*



Here we see that we need 3 layers to yield a receptive field that covers 8 units of inputs which is larger than the sequence length of 5. Note, z_t^L was used to denote the hidden variables where L represented the layer and that t represented the instance.

Similarly, a TCN requires 3 layers to process a *sequence of inputs* $x_1, \ldots, x_{10}$ *if it has **incremental dilation size 2 and kernel size 3***. This can be derived with



Overall, we are trying to solve for the problem of

$$L:\ \min\left(1+\sum_{i=0}^{L-1}\#\text{padding}_i\right)\geq \text{input sequence length}$$

where $\#\text{padding}_i$ is the amount of padding required for layer i. Since the amount of padding required for a **Causal_Conv1d** layer is $(\text{kernel}-1)$ dilation, we can rewrite the conditions as

$$L:\ \min(1+\textstyle\sum_{i=0}^{L-1}(\text{kernel}-1)\ \text{dilation}_i)\geq |\mathcal{X}|.$$

Now given that each image of pMNIST is of length $|\mathcal{X}|=784$, and that jerrybai1995 has decided to use TCNs with kernel size 7 with incremental dilation size 2, then the network would need 8 layers of **Causal_Conv1d** to yield an RF of 1531 to cover each image. If there were only 7 layers instead, then the RF would only be 763 < 784 and hence insufficient.