

برای افزایش دقت و بهبود عملکرد شبکه راه‌حل‌هایی مانند افزایش عمق شبکه (افزایش تعداد لایه‌ها) و یا افزایش عرض (افزودن تعداد نورون‌ها در یک لایه) به کار برده می‌شود که لزوماً راه‌حل خوبی نیستند. یک راه‌حل موثرتر افزایش cardinality است. cardinality به اندازه‌ی یک مجموعه از transformationها یا تبدیل‌ها گفته می‌شود. در این روش یک مجموعه از transformationها با توپولوژی یکسان روی یک نورون انجام می‌شود و نتایج مربوط به آن‌ها با هم جمع می‌شوند. بنابراین به غیر از ابعاد عمق و عرض شبکه بعد جدید دیگری به نام cardinality را معرفی کردیم که از اهمیت بالایی برخوردار است و می‌تواند از عمق و عرض موثرتر باشد. مقدار cardinality برای هر نورون می‌تواند متفاوت باشد.

وقتی از pooling استفاده می‌کنیم روی یک مجموعه از نورون‌ها یک transformation اعمال کرده و روی نتایج حاصل عمل ماکسیمم‌گیری یا میانگین‌گیری یا ... را حساب می‌کنیم اما وقتی از cardinality استفاده می‌کنیم روی یک نورون مجموعه‌ای از transformationها را اعمال کرده و بعد نتایج حاصل را با هم جمع می‌کنیم بنابراین استفاده از cardinality را می‌توان معادل اعمال pooling نه بر روی عرض یعنی چند نورون بلکه بر روی مجموعه‌ای از transformationها یعنی همین بعد جدیدی که از آن صحبت کردیم دانست.

شباهت‌ها:

لایه‌های کانولوشنی و لایه‌های بازگشتی هر دو دارای خاصیت اشتراک پارامتر (Parameter Sharing) هستند.

لایه‌های کانولوشنی و لایه‌های بازگشتی هر دو در معرض مشکلات ناپدید شدن گرادیان و انفجار گرادیان هستند.

تفاوت‌ها:

لایه‌های کانولوشنی	لایه‌های بازگشتی
برای داده‌های فضایی و grid مثل عکس مناسب است.	برای داده‌های زمانی و ترتیبی مناسب است.
لایه‌های کانولوشنی قدرتمندتر هستند.	لایه‌های بازگشتی سازگاری و ویژگی کمتری دارد.
لایه‌های کانولوشنی ورودی‌های اندازه ثابت را می‌گیرد و خروجی‌هایی با اندازه ثابت تولید می‌کند.	لایه‌های بازگشتی می‌تواند با طول ورودی / خروجی دلخواه کار کند
CNN نوعی شبکه عصبی مصنوعی feed-forward با انواعی مختلفی از پرسپترون‌های چند لایه است که طراحی شده تا از مقدار پیش‌پردازش بکاهد.	RNN برخلاف شبکه‌های عصبی feed-forward می‌توانند از حافظه داخلی‌شان برای پردازش دنباله‌های ورودی دلخواه استفاده کنند.
لایه‌های کانولوشنی از الگوی اتصال بین نورون‌ها استفاده می‌کنند.	لایه‌های بازگشتی از اطلاعات سری زمانی استفاده می‌کنند.
CNN برای پردازش تصاویر و فیلم مناسبتر است.	RNN ها برای تحلیل متن و گفتار مناسبتر هستند.

https://www.tutorialspoint.com/tensorflow/tensorflow_cnn_and_rnn_difference.htm

<https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/>

-3

برای هر دو مدل تعداد پارامترها به صورت زیر محاسبه می‌شود:

Number of parameters in SimpleRNN = $h\# * (h\# + x\# + \text{bias}\#)$

First SimpleRNN:

$$40 * (40 + 8 + 1) = 40 * 49 = 1960$$



h x bias

Second SimpleRNN:

$$50 * (50 + 40 + 1) = 50 * 91 = 4550$$

h x bias



Dense:

$$1 * (50 + 1) = 51$$

Total parameters = 1960 + 4550 + 51 = 6561

```
model_1 = tf.keras.models.Sequential([
    tf.keras.layers.Input(shape=(10,8)),
    tf.keras.layers.SimpleRNN(40, return_sequences=True),
    tf.keras.layers.SimpleRNN(50, return_sequences=True),
    tf.keras.layers.Dense(1)
])
model_1.summary()
```

Model: "sequential_5"

Layer (type)	Output Shape	Param #
=====		
simple_rnn_10 (SimpleRNN)	(None, 10, 40)	1960

simple_rnn_11 (SimpleRNN)	(None, 10, 50)	4550

dense_5 (Dense)	(None, 10, 1)	51
=====		
Total params: 6,561		
Trainable params: 6,561		
Non-trainable params: 0		

```
model_2 = tf.keras.models.Sequential([
    tf.keras.layers.Input(shape=(10,8)),
    tf.keras.layers.SimpleRNN(40, return_sequences=True),
    tf.keras.layers.SimpleRNN(50),
    tf.keras.layers.Dense(1)
])
model_2.summary()
```

Model: "sequential_6"

Layer (type)	Output Shape	Param #
simple_rnn_12 (SimpleRNN)	(None, 10, 40)	1960
simple_rnn_13 (SimpleRNN)	(None, 50)	4550
dense_6 (Dense)	(None, 1)	51
Total params: 6,561		
Trainable params: 6,561		
Non-trainable params: 0		

در هر دو مدل فوق در لایه اول SimpleRNN مقدار return_sequence برابر True است یعنی در دنباله‌ی خروجی دنباله‌ی کامل یعنی 10*40 برگردانده می‌شود. در مدل اول در لایه دوم SimpleRNN باز هم مقدار return_sequence برابر True است و در دنباله‌ی خروجی دنباله‌ی کامل (هر 10 خروجی) یعنی 10*50 برگردانده می‌شود (many to many) اما در مدل دوم در لایه دوم SimpleRNN مقدار return_sequence برابر مقدار پیش‌فرض یعنی False است و در دنباله خروجی فقط خروجی لایه‌ی آخر (یک خروجی) برگردانده می‌شود یعنی 50 (many to one).

-4

خیر نمی‌توان از این شبکه برای پیش‌بینی بر اساس 40 روز گذشته استفاده کرد چون در ورودی آن به صورت fix بیست روز را در نظر گرفتیم. تغییری که لازم است این است که shape ورودی را برابر (None,1) قرار دهیم.