KOMINFO digitalent

PRO Professional Academy

Digital Talent Scholarship 2022

Natural Language Processing Pt 2

Lead a sprint through the Machine Learning Track

Agenda

- More Word Embedding
- Sequence Model

Are your students ML-ready?

Apa yang kita pelajari kemarin?

Konsep Word Embedding

Manusia memiliki keunggulan dalam menata kata sampai dengan kalimat seperti pada contoh dibawah ini

- Kakak dari mama -> Tante
- Abang dari Mama -> Om

Namun berbeda dengan mesin, mesin tidak mengerti apa panggilan dari kakaknya mama dan lain sebagainya. Mesin hanya mengenal angka.

Apa itu Word Embedding

Merupakan representasi dari semantik dimana merupakan *de facto standard* pada NLP yang merupakan *word embeddings*, Vektor ini merepresentasikan :

- *Distributed*: informasi yang didistribusikan ke seluruh index
- *Distributional :* Informasi yang berasal dari distribusi kata dalam *corpus* (bagaimana bisa terjadi dalam teks)

Apa itu kata ?

Machine-Friendly Representation

Bagaimana cara kita merepresentasikan kata pada beberapa segment text agar *machine-friendly* adalah sebagai berikut ini:

1. Bag-of-words : tidak ada *word order*
2. Sequence of numerical indices : relatif tidak informatif
3. One hot vectors : ruang yang tidak efisien, *curse of dimensionality*
4. *Scores from lexicons, or hand-engineered features :* mahal dan tidak *scalable*

[Converting words to numbers, Word Embeddings - YouTube]

[MIT 6.S191 (2018): Sequence Modeling with Neural Networks - YouTube]

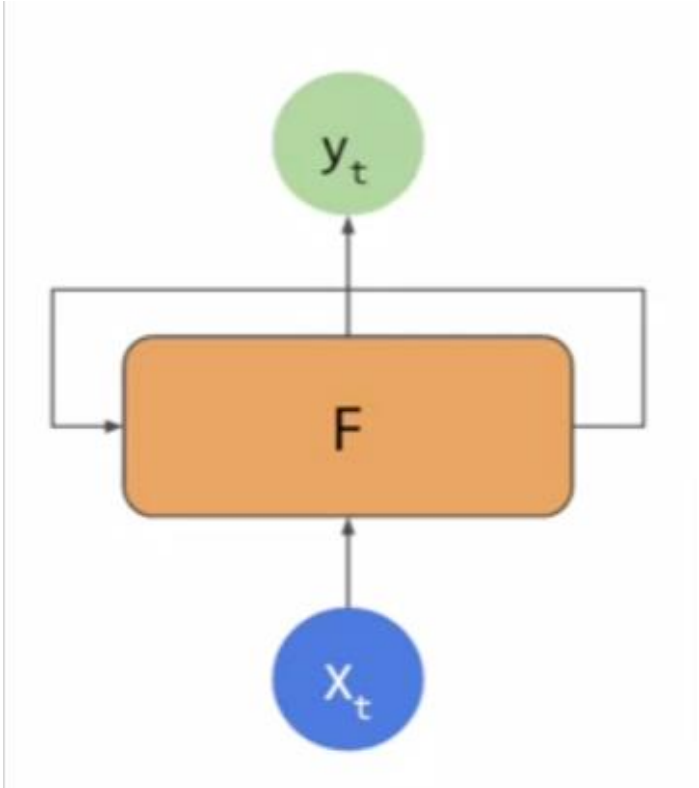
Demo IMDB Reviews Subword

RNN

[Recurrent Neural Networks - YouTube]

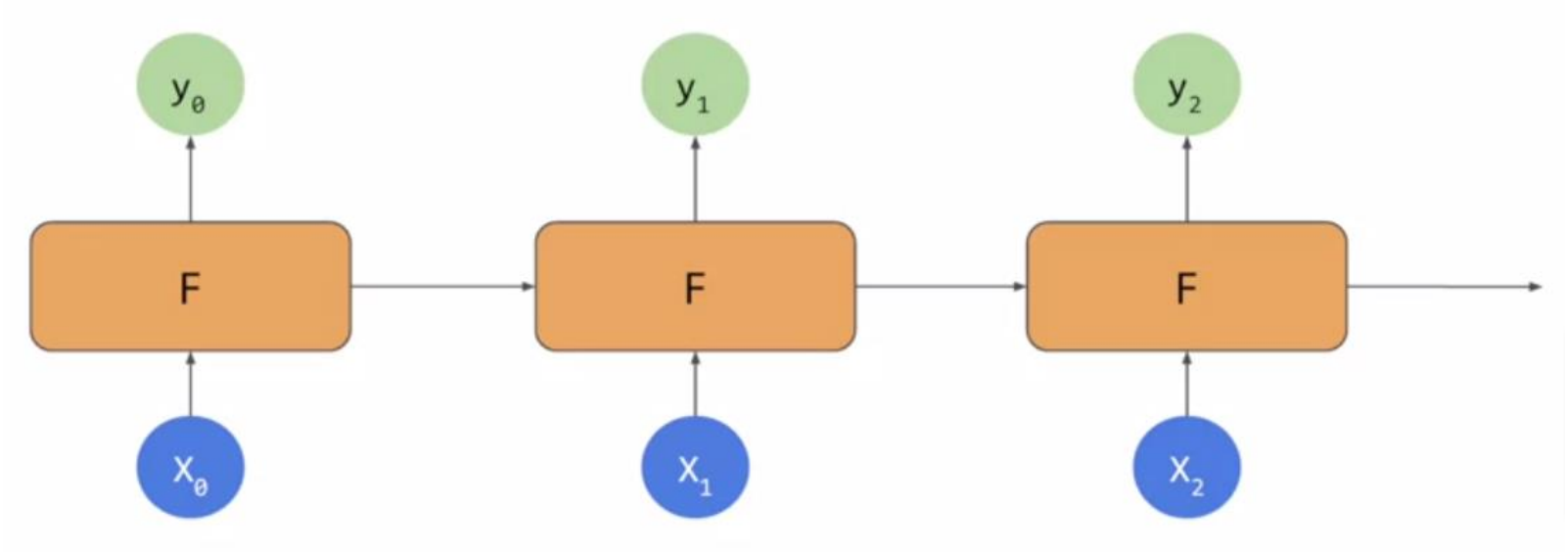
RNN

Recurrent Neural Network, atau yang biasa disingkat menjadi **RNN**, merupakan salah satu bentuk arsitektur Artificial Neural Networks (ANN) yang dirancang khusus untuk memproses data yang bersambung/ berurutan (sequential data). RNN biasanya digunakan untuk menyelesaikan tugas yang terkait dengan data time series, misalnya data ramalan cuaca.



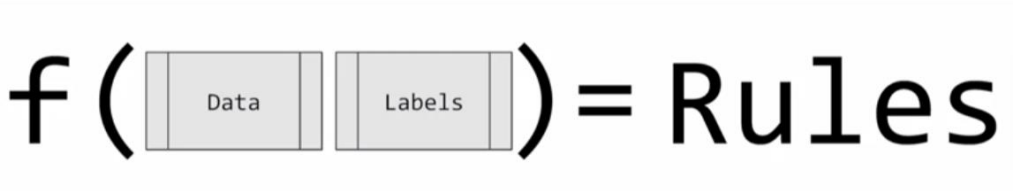
RNN

Visualisasi RNN



Kelemahan DNN tanpa RNN

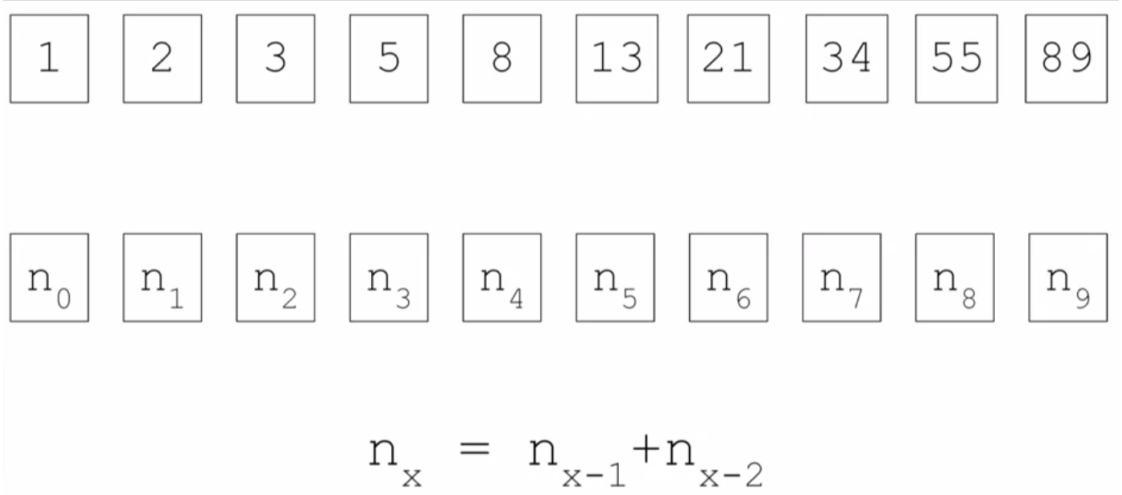
1. Tidak ada konteks dalam subwords
2. Sequence menjadi sangat penting dalam mengerti arti sebuah kata.



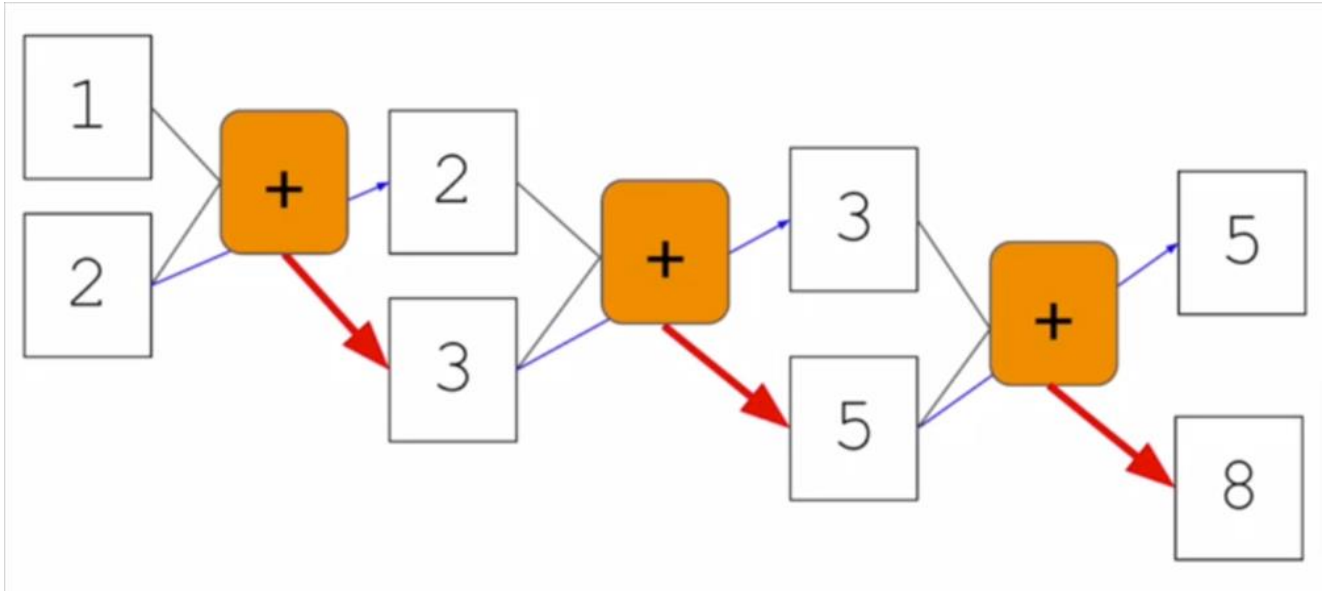
Tidak ada sequence dalam DNN biasa. Maka dari itu kita memerlukan **RNN**.

Contoh RNN

Fibonacci, adalah deret angka yang diperoleh dengan menjumlahkan dua angka sebelumnya. Artinya Fibonacci sangat mementingkan **sequence** alias **urutan**.



Visualisasi Fibonacci



LSTM and GRU

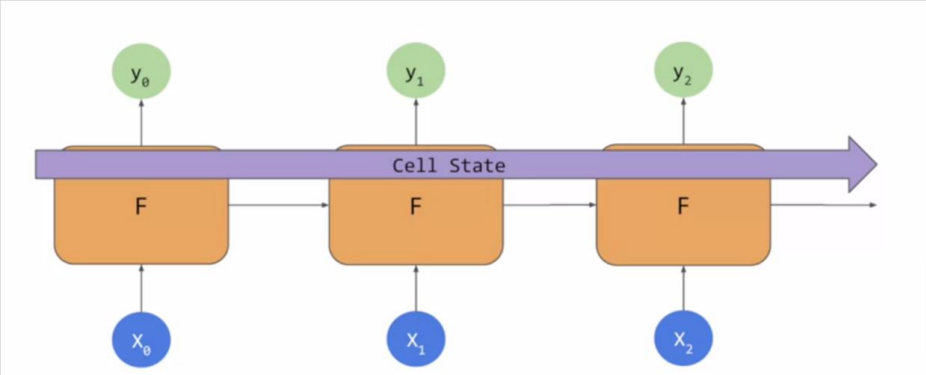
[Illustrated Guide to LSTM's and GRU's - YouTube]

LSTM

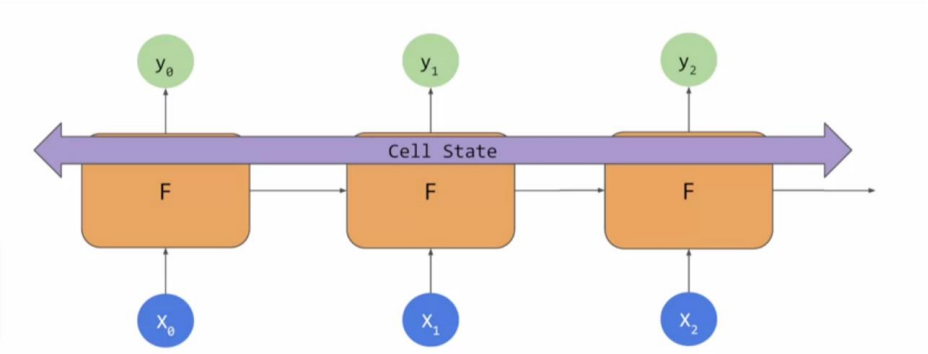
Long Short-Term Memory (LSTM), adalah salah satu jenis dari Recurrent Neural Network (RNN) dimana dilakukan modifikasi pada RNN dengan menambahkan memory cell yang dapat menyimpan informasi untuk jangka waktu yang lama.

LSTM

Directional



Bidirectional



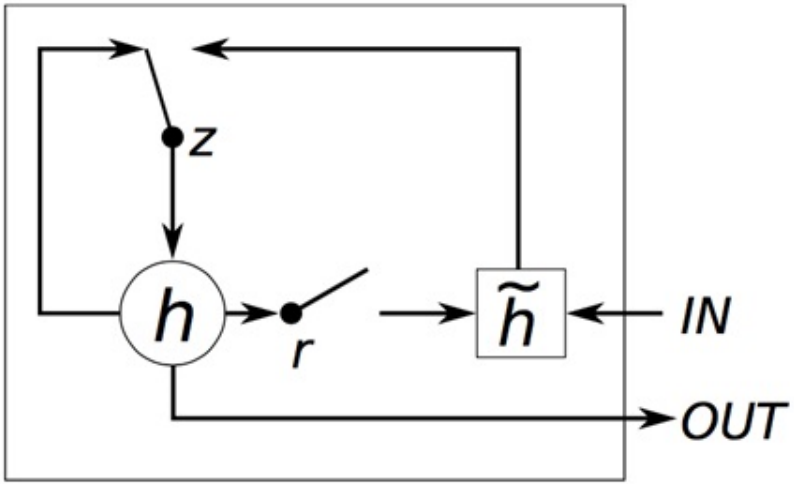
LSTM

```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(tokenizer.vocab_size, 64),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(tokenizer.vocab_size, 64),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64, return_sequences=True)),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(32)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

GRU

Gated Recurrent Unit, salah satu varian LSTM yang dapat membuat setiap recurrent unit untuk dapat menangkap dependencies dalam skala waktu yang berbeda-beda secara adaptif.



GRU

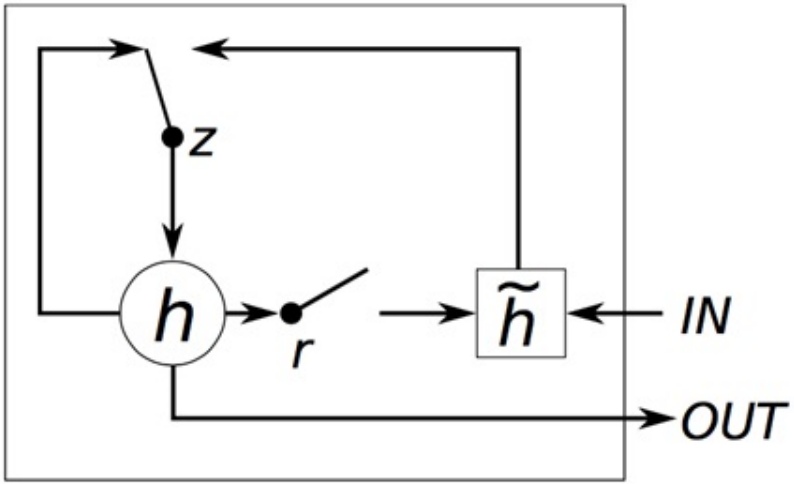
Manusia tidak memerlukan semua informasi masa lalu untuk membuat keputusan.

Ibarat saat kita beli sarapan pagi ini, kita tidak perlu membuat keputusan untuk membeli makanan apa berdasarkan baju apa yang kita pakai 7 hari yang lalu. Dengan begitu GRU membatasi informasi yang digunakan RNN.

GRU

Reset Gate - Menggabungkan input baru dengan informasi lalu **(r)**

Update Gate - Menunjukkan berapa banyak informasi masa lalu yang harus disimpan **(z)**



KOMINFO digitalent

GRU

```
model_gru = tf.keras.Sequential([
    tf.keras.layers.Embedding(vocab_size, embedding_dim, input_length=max_length),
    tf.keras.layers.Bidirectional(tf.keras.layers.GRU(gru_dim)),
    tf.keras.layers.Dense(dense_dim, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Perbedaan LSTM dan GRU

Perbedaan utama antara GRU dan LSTM adalah bahwa GRU memiliki dua gerbang yang di-**reset** dan di-**update**, sementara LSTM memiliki tiga gerbang yaitu **input**, **output**, **forget**. GRU lebih simple daripada LSTM karena memiliki jumlah gerbang yang lebih sedikit.

Jika dataset kecil maka GRU lebih cocok. LSTM disarankan untuk dataset yang lebih besar.

Q & A

Thank You