Taller de Aprendizaje Automático

Taller 8:

Predicciones con Redes Neuronales Recurrentes (RNN)

Instituto de Ingeniería Eléctrica
Facultad de Ingeniería

UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Montevideo, 2024

Tabla de contenido

Objetivos del Taller 8

- Manipular secuencias de datos.
- Comparar diferentes enfoques de modelos de RNN para un problema concreto.

El problema: Demanda de Bicicletas

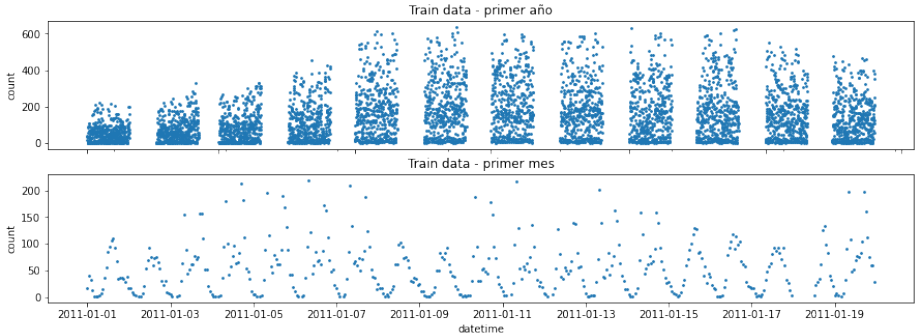


Figure: Desafío de Kaggle de Demanda de Bicicletas

Manipulación de secuencias de datos

- Datos Faltantes
  Se deben rellenar para que todas las secuencias correspondan al mismo intervalo temporal.

```
1 df_train.loc[45:50]
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
45	2011-01-02 22:00:00	1	0	0	1	9.84	10.605	44	19.9995	0	9	9
46	2011-01-02 23:00:00	1	0	0	1	9.02	11.365	47	11.0014	0	8	8
47	2011-01-03 00:00:00	1	0	1	1	9.02	9.850	44	23.9994	0	5	5
48	2011-01-03 01:00:00	1	0	1	1	8.20	8.335	44	27.9993	0	2	2
49	2011-01-03 04:00:00	1	0	1	1	6.56	6.820	47	26.0027	0	1	1
50	2011-01-03 05:00:00	1	0	1	1	6.56	6.820	47	19.0012	0	3	3

Figure: Datos Faltantes.

Manipulación de secuencias de datos

- Datos Faltantes

```
def FilledIn(df):
    df_aux = df.copy()
    df_out = pd.DataFrame(columns=df_aux.columns)
    df_aux['datetime'] = pd.to_datetime(df_aux['datetime'])
    df_aux = df_aux.set_index('datetime')
```

Manipulación de secuencias de datos

- Datos Faltantes

```
def FilledIn(df):
    df_aux = df.copy()
    df_out = pd.DataFrame(columns=df_aux.columns)
    df_aux['datetime'] = pd.to_datetime(df_aux['datetime'])
    df_aux = df_aux.set_index('datetime')
    for year in [2011, 2012]:
      for month in range(12):
        start_date = datetime(year, month+1, 1, 0, 0, 0)
        last_day_of_month = calendar.monthrange(year, month+1)[1]
        end_date = datetime(year, month+1, last_day_of_month, 23, 0, 0)
        # Se agregan las marcas de tiempo que faltan
        df_month = df_aux[start_date:end_date]
        df_month = df_month.resample('H').asfreq()
```

Manipulación de secuencias de datos

- Datos Faltantes

```
def FilledIn(df):
    df_aux = df.copy()
    df_out = pd.DataFrame(columns=df_aux.columns)
    df_aux['datetime'] = pd.to_datetime(df_aux['datetime'])
    df_aux = df_aux.set_index('datetime')
    for year in [2011, 2012]:
      for month in range(12):
        start_date = datetime(year, month+1, 1, 0, 0, 0)
        last_day_of_month = calendar.monthrange(year, month+1)[1]
        end_date = datetime(year, month+1, last_day_of_month, 23, 0, 0)
        # Se agregan las marcas de tiempo que faltan
        df_month = df_aux[start_date:end_date]
        df_month = df_month.resample('H').asfreq()
        # Rellenar los datos faltantes===========
          #Su código
        #=====================================
        df_month = df_month.reset_index()
        df_out = df_out.append(df_month)
    df_out = df_out.reset_index(drop=True)
    return df_out
```

Manipulación de secuencias de datos

- Ingeniería y Estandarización de los datos.

```
class TimeFeatures(BaseEstimator, TransformerMixin):
  def __init__(self):
    self
  def fit(self, X, y=None):
    # X debe ser un DataFrame
    return self
  def transform(self, X):
    X_aux = X.copy()
    X_aux['datetime'] = pd.to_datetime(X_aux['datetime'])
    X_aux['month'] = X_aux['datetime'].dt.month
    X_aux['weekday'] = X_aux['datetime'].dt.weekday
    X_aux['hour'] = X_aux['datetime'].dt.hour
    X_aux = X_aux.drop('datetime', axis=1)
    return X_aux

cat_features = ['season', 'weather', 'month', 'weekday', 'hour']
num_features = ['temp', 'atemp', 'humidity', 'windspeed']
# [holiday, workingday] ya son onehot
scaler = ColumnTransformer([('cat', OneHotEncoder(), cat_features),
                            ('num', StandardScaler(), num_features),
                            ], remainder='passthrough')
```

Manipulación de secuencias de datos

- TF Dataset de Secuencias.
  - Predecir la demanda de la próxima hora, dado que se conocen los datos de las últimas 24 horas.

Manipulación de secuencias de datos

- TF Dataset de Secuencias.
  - Predecir la demanda de la próxima hora, dado que se conocen los datos de las últimas 24 horas.
  - Se sugiere utilizar la función keras.preprocessing.timeseries_dataset_from_array().

```
tf.keras.preprocessing.timeseries_dataset_from_array(
    data, targets, sequence_length, sequence_stride=1, sampling_rate=1,
    batch_size=128, shuffle=False, seed=None, start_index=None, end_index=None
)
```

Manipulación de secuencias de datos

- TF Dataset de Secuencias.
  - Predecir la demanda de la próxima hora, dado que se conocen los datos de las últimas 24 horas.
  - Se sugiere utilizar la función keras.preprocessing.timeseries_dataset_from_array().

```
tf.keras.preprocessing.timeseries_dataset_from_array(
    data, targets, sequence_length, sequence_stride=1, sampling_rate=1,
    batch_size=128, shuffle=False, seed=None, start_index=None, end_index=None
)
```

  - Se debe tener en cuenta el salto temporal debido a los datos de test.

Manipulación de secuencias de datos

- TF Dataset de Secuencias.
  - Predecir la demanda de la próxima hora, dado que se conocen los datos de las últimas 24 horas.
  - Se sugiere utilizar la función keras.preprocessing.timeseries_dataset_from_array().

```
tf.keras.preprocessing.timeseries_dataset_from_array(
    data, targets, sequence_length, sequence_stride=1, sampling_rate=1,
    batch_size=128, shuffle=False, seed=None, start_index=None, end_index=None
)
```

  - Se debe tener en cuenta el salto temporal debido a los datos de test.
  - Se recomienda tomar como conjunto de train los primeros 15 días del mes y como conjunto de validación los días del 16 al 19.

Modelos

- Naive Forecasting: Baseline, predice un valor como el valor del dato anterior.

Modelos

- Naive Forecasting: Baseline, predice un valor como el valor del dato anterior.
- RNNs:

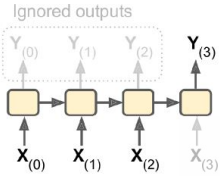
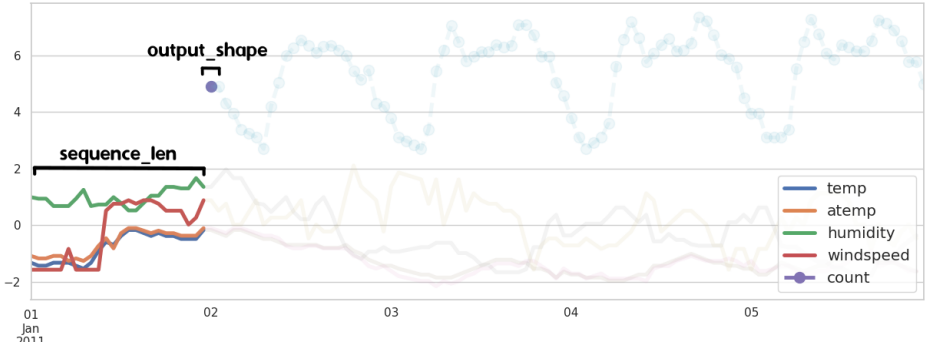
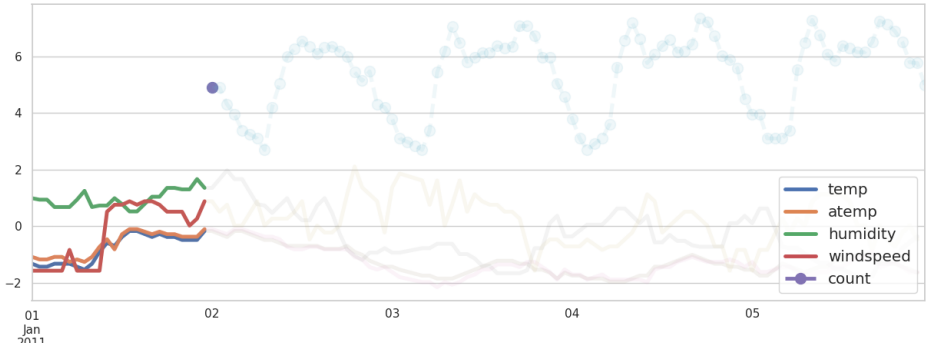


Figure: Seq-to-Vector

Seq-to-Vec


output_shape
sequence_len
6
4
2
0
−2
01
Jan
2011
02
03
04
05
temp
atemp
humidity
windspeed
count

Seq-to-Vec



Modelos

- Naive Forecasting: Baseline, predice un valor como el valor del dato anterior.
- RNNs:

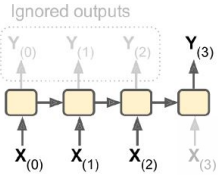


Figure: Seq-to-Vector

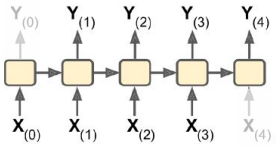
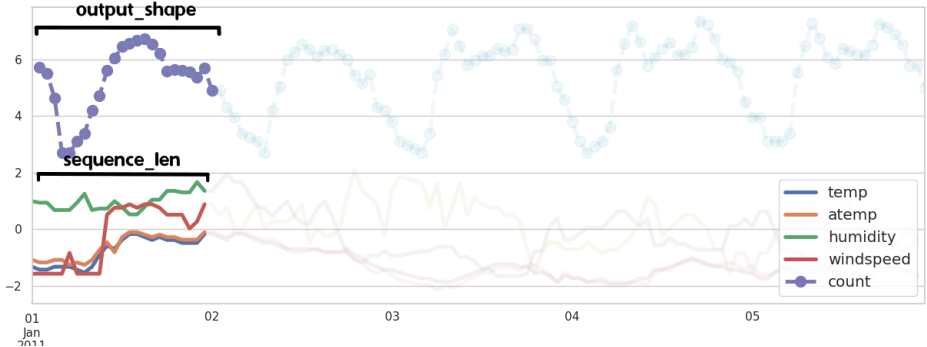
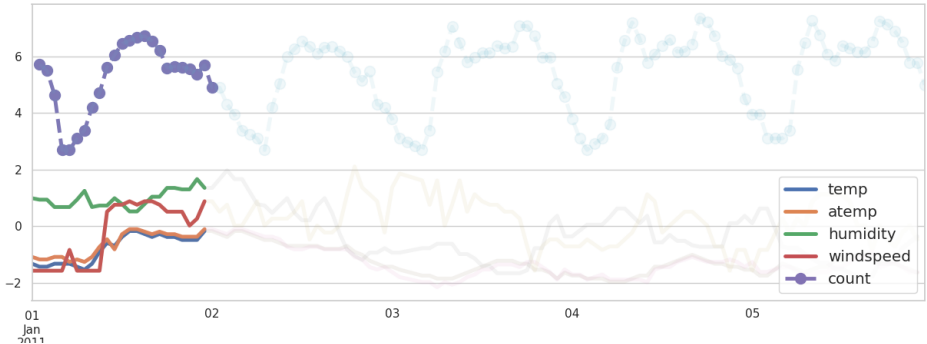


Figure: Seq-to-Seq

Seq-to-Seq


output_shape
sequence_len
6
4
2
0
−2
01
Jan
2011
02
03
04
05
temp
atemp
humidity
windspeed
count

Seq-to-Seq


6
4
2
0
−2
01
Jan
2011
02
03
04
05
temp
atemp
humidity
windspeed
count

Modelos

- Naive Forecasting: Baseline, predice un valor como el valor del dato anterior.
- RNNs:

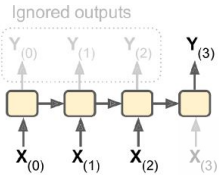


Figure: Seq-to-Vector

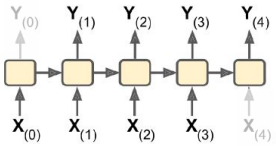


Figure: Seq-to-Seq

- Capas SimpleRNN y LSTM

Modelos

- Naive Forecasting: Baseline, predice un valor como el valor del dato anterior.
- RNNs:

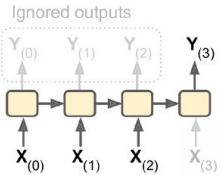


Figure: Seq-to-Vector

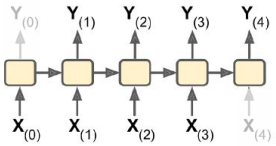


Figure: Seq-to-Seq

- Capas SimpleRNN y LSTM
- Diferentes largos de secuencia
- (Opcional) Regularización