SOURCE-AGNOSTIC GRAVITATIONAL-WAVE DETECTION WITH TRANSFORMERS

NATNAEL BERHANE DEBRU (KHALIFA UNIVERSITY) – EP-CMG-OS

SUPERVISORS: NADYA CHERNYAVSKAYA, MAURIZIO PIERINI

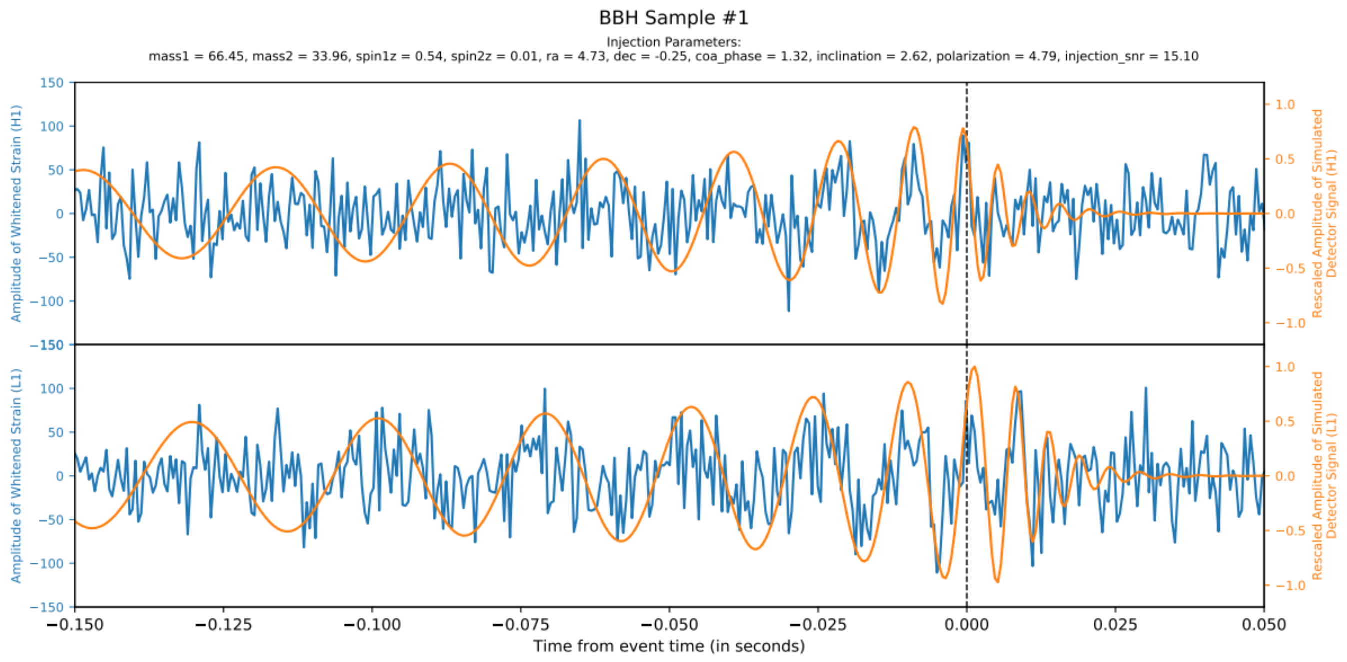
Summer Student Session 2022: August 3, 2022

Machine Learning for Particle Physics (mPP)

- CERN R&D project to investigate and promote applications of modern machine learning (ML) to particle physics
- Developing and deploying ML solutions for CERN experiments, such as trigger, event reconstruction, simulation, etc.
- Interest in applying ML techniques to anomalous signal detection in gravitational wave (GW) detectors (LIGO)
- E. A. Moreno et al., “Source-agnostic gravitational-wave detection with recurrent autoencoders”

  (https://arxiv.org/pdf/2107.12698.pdf)

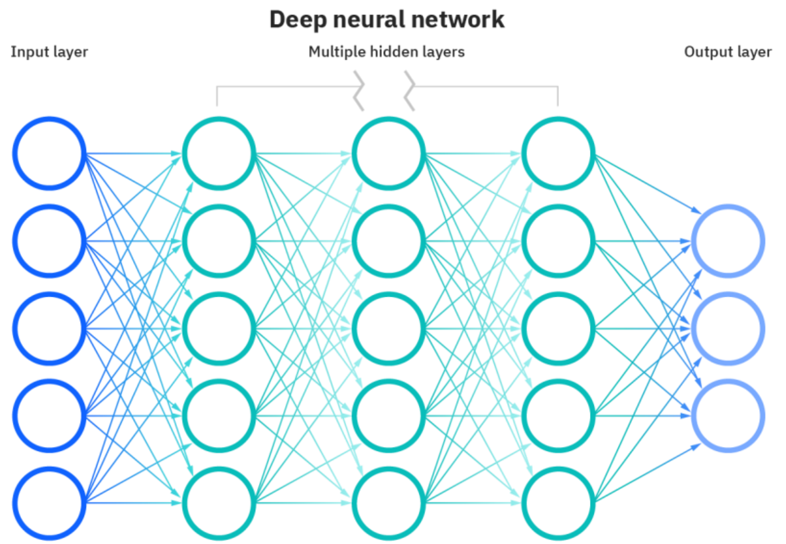
GW Detection at LIGO

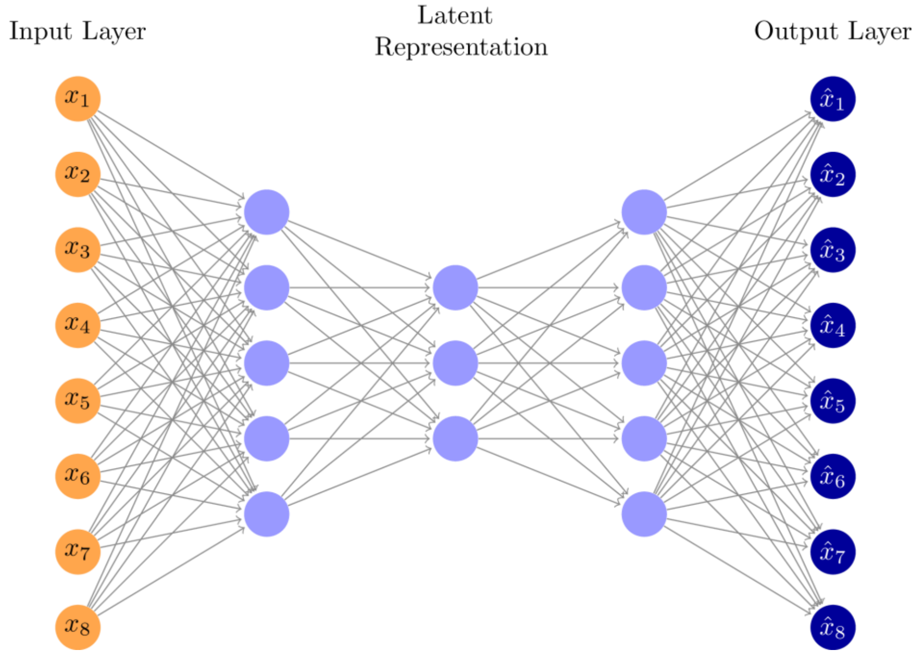


- Matched filtering: matching data with simulated signal templates. Used to identify signal as well as determine event parameters
- ML techniques: models trained with simulated data to extract signals from noisy data. Result in increased accuracy
- Drawback: rely on pre-defined signals, cannot identify unmodeled signals or unknown signals from exotic sources

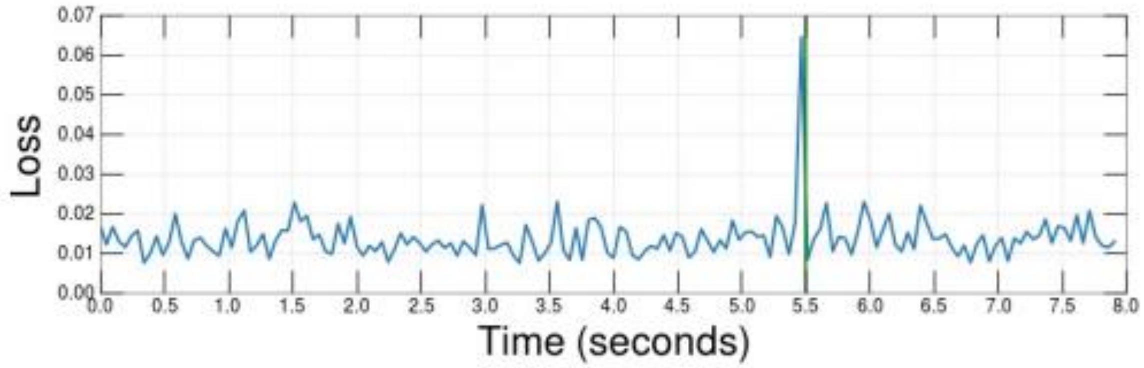
Source: https://arxiv.org/pdf/2107.12698.pdf

Deep Learning for GW Signal Detection

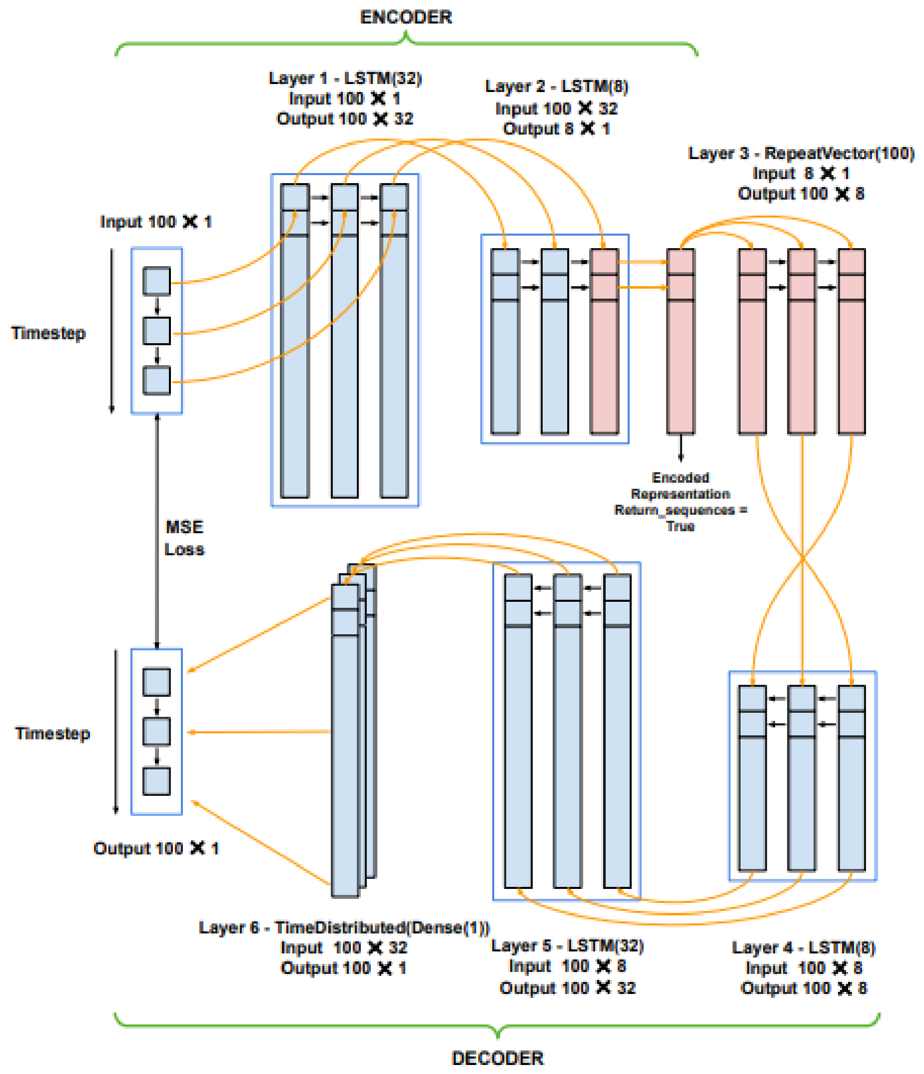




- Deep learning (DL) with artificial neural networks (ANNs)
- Supervised vs Unsupervised learning
- Autoencoders (AEs) for anomaly detection:
  - Compression-decompression algorithm
  - Unsupervised learning (sensitivity to unmodeled signals)
  - Can be trained directly with GW data



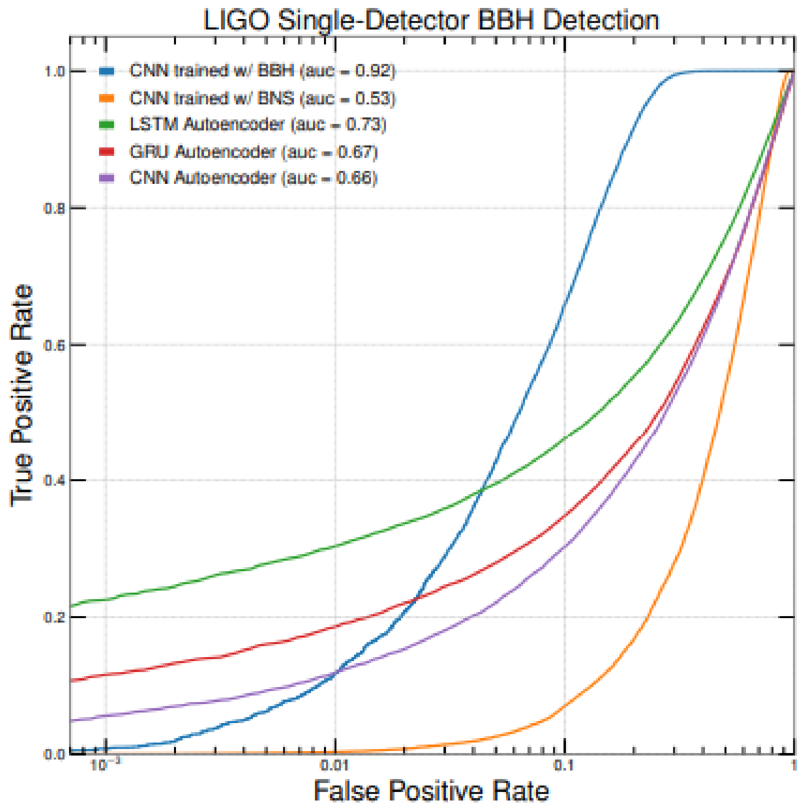
Previous Work



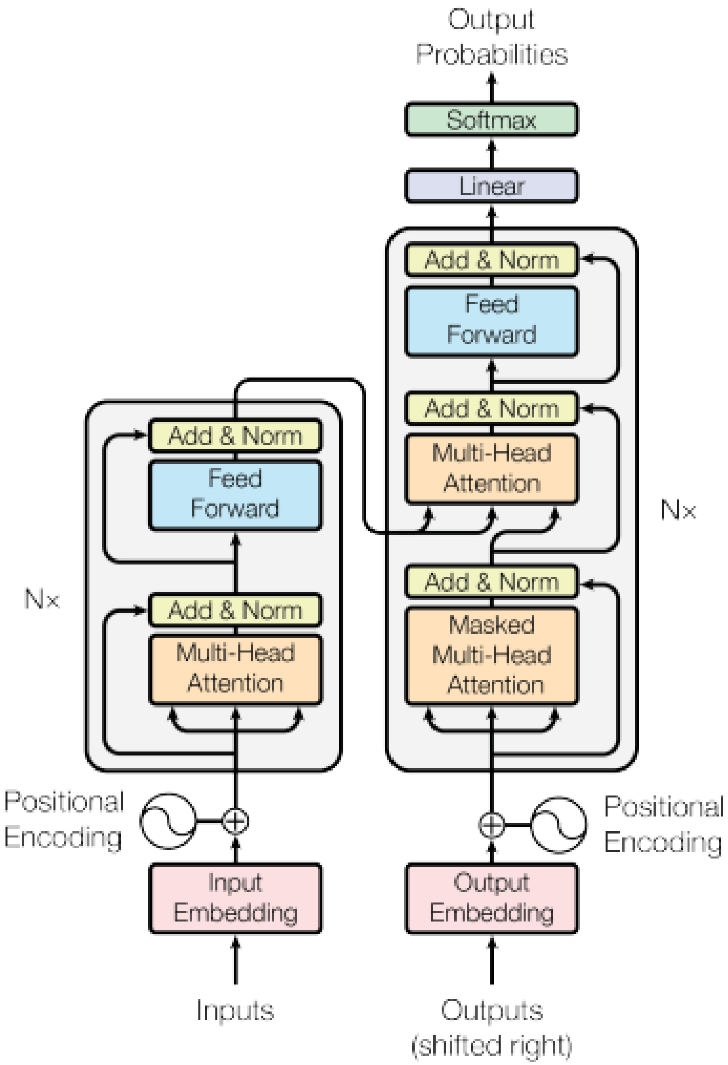
Source: https://arxiv.org/pdf/2107.12698.pdf

- Architecture: Recurrent (LSTM, GRU) and Convolutional (CNN) autoencoders

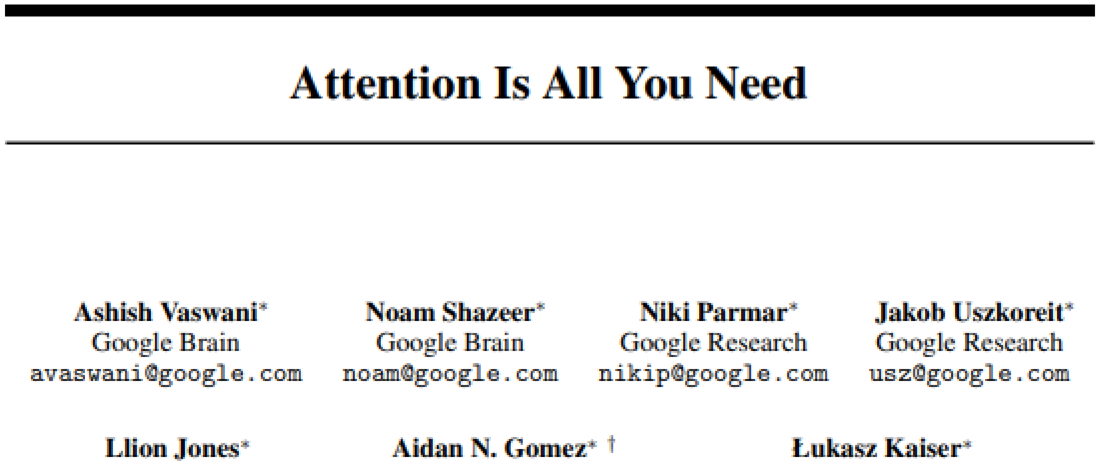
- Results show recurrent autoencoders are efficient at anomaly detection.
- LSTM model performs best



Transformers



- Recent sequence to sequence model, first developed for text-based language translations (NLP)
- Primarily relies on attention layer
- Demonstrated efficiency at predicting future sequences once trained with input and corresponding output data (supervised)
- Particularly suitable for time-series data → suitable candidate for GW anomaly detection



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones*

Aidan N. Gomez* †

Łukasz Kaiser*

Source: https://arxiv.org/pdf/1706.03762.pdf

Transformers for GW Anomaly Detection

```python
def transformer_encoded_autoencoder (X):  #X input data with shape [samples,timestep,features]
    """Autoencoder with a transformer layer encoder and convolution layer decoder"""
    #Transformer hyperparameters
    num_heads=5  #number of multi head attention layers

    #encoder
    inputs = Input(shape=(X.shape[1],X.shape[2]))
    L1 = transformer_encoder(inputs,head_size=100,num_heads=num_heads,ff_dim=800)
    L2 = transformer_encoder(L1,head_size=100,num_heads=num_heads,ff_dim=200)
    L3 = transformer_encoder(L2,head_size=100,num_heads=num_heads,ff_dim=50)
    #decoder
    L4 = Conv1D(128, 3, activation='relu', padding="same")(L3)
    output = Conv1D(1, 3, activation='relu', padding="same")(L4)

    model = Model(inputs=inputs, outputs = output)
    return model
```

```python
def transformer_encoder(inputs, head_size, num_heads, ff_dim, dropout=0):
    # Normalization and Attention
    x = layers.LayerNormalization(epsilon=1e-6)(inputs)
    x = layers.MultiHeadAttention(
        key_dim=head_size, num_heads=num_heads, dropout=dropout
    )(x, x)
    x = layers.Dropout(dropout)(x)
    res = x + inputs

    # Feed Forward Part
    x = layers.LayerNormalization(epsilon=1e-6)(res)
    x = layers.Conv1D(filters=ff_dim, kernel_size=1, activation="relu")(x)
    x = layers.Dropout(dropout)(x)
    x = layers.Conv1D(filters=inputs.shape[-1], kernel_size=1)(x)
    return x + res
```

Source: https://keras.io/examples/timeseries/timeseries_transformer_classification/

Preliminary model:

- Transformer in an unsupervised autoencoder architecture
- Autoencoder with transformer layers as encoder and convolution layers as decoder
- Convolution layers used instead of original feed-forward layers in order to preserve order of time-series data
- Trained with simulated LIGO noise data

Next Steps

- Evaluation of preliminary model
- Hyperparameter optimization: optimizing model architecture to obtain optimum performance
- Optimization of data processing techniques (for ex., number of time steps used for a single event, data augmentation process)
- Improvement of dual-detector anomaly detection system (combining data from both LIGO detectors to make more accurate detections)
- Development and testing of other transformer-based architectures:
  - "Traditional" transformer trained to predict future time-series (such as GW data) based on past data

THANK YOU!

ANY QUESTIONS?

Contact:
natnael.berhane.debru@cern.ch
100049891@ku.ac.ae

جامعة خليفة
Khalifa University