LAB 10 | 10 المخبر

Load Forecasting with a Transformer

التنبؤ بالحمل الكهربائي باستخدام Transformer

MATLAB R2023b+ practical laboratory using self-attention and positional embeddings

مخبر عملي باستخدام الانتباه الذاتي والترميز الموضعي

Prepared by | إعداد: Dr.-Ing. Aouss Gabash

🎯 Laboratory Objectives

- Prepare hourly load sequences for transformer regression.
- Create feature embeddings and positional embeddings.
- Use multi-head self-attention in MATLAB.
- Train a one-step-ahead forecasting model.
- Compare transformer, persistence, and optionally LSTM results.
- Visualize attention-compatible temporal behavior and errors.

🎯 أهداف المخبر

- تجهيز متتاليات حمل ساعية للانحدار.
- إنشاء تمثيل للميزات والمواضع.
- استخدام الانتباه الذاتي متعدد الرؤوس في MATLAB.
- تدريب نموذج لتوقع الساعة التالية.
- مقارنته بخط الأساس وLSTM اختياريًا.
- تحليل النتائج والأخطاء.

1. Software Requirement

Use MATLAB R2023b or newer with Deep Learning Toolbox because this laboratory uses `selfAttentionLayer` and `positionEmbeddingLayer`.

1. متطلبات البرنامج

يتطلب المخبر MATLAB R2023b أو أحدث مع Deep Learning Toolbox.

2. Forecasting Task

$$[P_{t-47}, \dots, P_t] \rightarrow P_{t+1}$$

48-hour window **One-step ahead** **Regression**

2. مسألة التنبؤ

نستخدم آخر 48 ساعة لتوقع حمل الساعة التالية، مما يسمح للنموذج بمشاهدة نمط يومين متتاليين.

3. Complete MATLAB Script

```
%% Lab 10: Transformer Load Forecasting
% MATLAB R2023b or newer
clear; clc; close all;
rng(10);

%% 1) Generate hourly load data
nDays = 240;
N = 24*nDays;
t = (0:N-1)';
hour = mod(t,24);
day = floor(t/24);
dow = mod(day,7);

temperature = 17 ...
    + 9*sin(2*pi*(hour-14)/24) ...
    + 6*sin(2*pi*day/240) ...
    + 1.5*randn(N,1);

daily = 115 ...
    + 24*sin(2*pi*(hour-7)/24) ...
    + 9*sin(4*pi*(hour-5)/24);

weekly = 9*(dow < 5) - 8*(dow >= 5);
seasonal = 10*sin(2*pi*day/240);

loadMW = daily + weekly + seasonal ...
    + 0.8*max(temperature-23,0) ...
    + 0.45*max(11-temperature,0) ...
    + 2.2*randn(N,1);

%% 2) Create sequence-to-one samples
seqLen = 48;
numSamples = N-seqLen;

X = cell(numSamples,1);
Y = zeros(numSamples,1);

for k = 1:numSamples
    X{k} = loadMW(k:k+seqLen-1)';   % 1 x 48
    Y(k) = loadMW(k+seqLen);           % next hour
end

%% 3) Chronological split
nTrain = floor(0.70*numSamples);
nVal   = floor(0.15*numSamples);
```

```
iTrain = 1:nTrain;
iVal   = nTrain+1:nTrain+nVal;
iTest  = nTrain+nVal+1:numSamples;

XTrain = X(iTrain); YTrain = Y(iTrain);
XVal   = X(iVal);   YVal   = Y(iVal);
XTest  = X(iTest);  YTest  = Y(iTest);

%% 4) Normalize using training data only
trainMatrix = cell2mat(XTrain);
mu = mean(trainMatrix,'all');
sigma = std(trainMatrix,0,'all');

XTrainN = cellfun(@(x)(x-mu)/sigma,XTrain, ...
    'UniformOutput',false);
XValN = cellfun(@(x)(x-mu)/sigma,XVal, ...
    'UniformOutput',false);
XTestN = cellfun(@(x)(x-mu)/sigma,XTest, ...
    'UniformOutput',false);

YTrainN = (YTrain-mu)/sigma;
YValN   = (YVal-mu)/sigma;

%% 5) Transformer architecture
numFeatures = 1;
embedDim = 32;
numHeads = 4;
ffnDim = 64;

layers = [
    sequenceInputLayer(numFeatures,Name="input")
    fullyConnectedLayer(embedDim,Name="projection")
    positionEmbeddingLayer(embedDim,seqLen,Name="position")
    additionLayer(2,Name="add_position")

    selfAttentionLayer(numHeads,embedDim, ...
        DropoutProbability=0.10,Name="attention")
    additionLayer(2,Name="add_attention")
    layerNormalizationLayer(Name="norm1")

    fullyConnectedLayer(ffnDim,Name="ffn1")
    reluLayer(Name="relu")
    dropoutLayer(0.10,Name="dropout")
    fullyConnectedLayer(embedDim,Name="ffn2")
```

```
    additionLayer(2,Name="add_ffn")
    layerNormalizationLayer(Name="norm2")

    globalAveragePooling1dLayer(Name="pool")
    fullyConnectedLayer(1,Name="forecast")
    regressionLayer(Name="output")
];

lgraph = layerGraph(layers);

% Residual and positional connections
lgraph = connectLayers(lgraph,"projection","add_position/in2");
lgraph = connectLayers(lgraph,"add_position","add_attention/in2");
lgraph = connectLayers(lgraph,"norm1","add_ffn/in2");

%% 6) Training options
options = trainingOptions("adam", ...
    MaxEpochs=70, ...
    MiniBatchSize=64, ...
    InitialLearnRate=1e-3, ...
    GradientThreshold=1, ...
    Shuffle="never", ...
    ValidationData={XValN,YValN}, ...
    ValidationFrequency=30, ...
    ValidationPatience=8, ...
    Verbose=false, ...
    Plots="training-progress");

%% 7) Train network
net = trainNetwork(XTrainN,YTrainN,lgraph,options);

%% 8) Test prediction
YPredN = predict(net,XTestN,MiniBatchSize=64);
YPred = YPredN*sigma + mu;

%% 9) Persistence baseline
YPersistence = cellfun(@(x)x(end),XTest);

%% 10) Performance metrics
metric = @(yp,yt) [ ...
    mean(abs(yp-yt)), ...
    sqrt(mean((yp-yt).^2)), ...
    100*mean(abs((yp-yt)./yt)) ];

mBase = metric(YPersistence,YTest);
```

```
mTrans = metric(YPred,YTest);

Results = array2table([mBase;mTrans], ...
    VariableNames={"MAE_MW","RMSE_MW","MAPE_percent"}, ...
    RowNames={"Persistence","Transformer"});

disp(Results);

%% 11) Plot forecasts
nShow = min(240,numel(YTest));
figure;
plot(YTest(1:nShow),"k",LineWidth=1.35); hold on;
plot(YPersistence(1:nShow),"--",LineWidth=1.0);
plot(YPred(1:nShow),LineWidth=1.2);
grid on;
xlabel("Test Hour");
ylabel("Load (MW)");
legend("Actual","Persistence","Transformer", ...
    Location="best");
title("Transformer One-Step-Ahead Load Forecast");

%% 12) Error analysis
e = YPred-YTest;

figure;
histogram(e,30);
grid on;
xlabel("Prediction Error (MW)");
ylabel("Frequency");
title("Transformer Error Distribution");

figure;
scatter(YTest,YPred,10,"filled");
grid on; axis equal;
xlabel("Actual Load (MW)");
ylabel("Predicted Load (MW)");
title("Actual versus Predicted Load");
```

3. الكود الكامل

ينشئ الكود بيانات حمل، يبني نوافذ بطول 48 ساعة، يطبّعها اعتمادًا على التدريب فقط، ثم يبني Encoder صغيرًا بانتباه متعدد الرؤوس.

4. Network Structure

1×48 load sequence → Projection to 32 channels → Position embedding →

4-head attention → FFN + pooling → Next-hour load

4. بنية الشبكة

تُرفع الإشارة من ميزة واحدة إلى 32 قناة، ثم يضاف الترميز الموضعي، وبعد الانتباه والشبكة الأمامية يجمع المتوسط الزمني التمثيل لإنتاج التوقع.

5. Residual Connections

$$y = LayerNorm(x + Attention(x))$$

$$z = LayerNorm(y + FFN(y))$$

The MATLAB graph explicitly connects the block input to the second input of each addition layer.

5. الوصلات المتبقية

تُجمع إشارة الدخل مع خرج الانتباه، ثم يُجمع خرج المرحلة الأولى مع خرج الشبكة الأمامية. تساعد هذه الوصلات على استقرار التدريب.

6. Important Hyperparameters

Parameter	Value	Meaning
Sequence length	48	Two days of hourly history
Embedding dimension	32	Internal feature space
Attention heads	4	Four relationship subspaces
FFN dimension	64	Nonlinear processing capacity
Dropout	0.10	Regularization

6. المعاملات المهمة

يمكن تغيير طول النافذة وبعد التمثيل وعدد الرؤوس وحجم الشبكة الأمامية ونسبة الإسقاط لدراسة أثرها في الدقة والكلفة.

7. Expected Behavior

- The transformer should learn the daily load cycle.
- It should usually outperform persistence when the load is changing.
- Validation loss should stop improving before training loss if overfitting begins.
- Peak hours may remain harder to predict.

A transformer is not automatically better than LSTM. Performance depends on data volume, window length, features, and tuning.

7. السلوك المتوقع

يُفترض أن يتعلم النموذج النمط اليومي، لكن التفوق على LSTM ليس مضمونًا، بل يعتمد على كمية البيانات والميزات والإعدادات.

8. Student Experiments

1. Test sequence lengths 24, 48, 72, and 168.
2. Test 1, 2, 4, and 8 attention heads.
3. Change embedding dimension from 32 to 64.
4. Add temperature and hour-of-day features.
5. Replace global average pooling with last-step indexing.
6. Compare with Lab 09 LSTM using identical data and splits.

8. تجارب الطالب

1. جرّب نوافذ 24 و48 و72 و168 ساعة.
2. جرّب 1 و2 و4 و8 رؤوس.
3. غيّر بعد التمثيل من 32 إلى 64.
4. أضف الحرارة والساعة كميزات.
5. قارن التجميع المتوسط باستخدام آخر خطوة.
6. قارن النتائج مباشرة مع LSTM في المخبر 09.

9. Mini Project

Day-ahead forecasting: Modify the model to output the next 24 hourly loads. The final fully connected layer must output 24 values, and each target must be a 24-element vector.

9. مشروع مصغر

حوّل النموذج من توقع ساعة واحدة إلى توقع اليوم التالي كاملًا، بحيث ينتج 24 قيمة حمل دفعة واحدة.

10. Laboratory Report

- Explain Q, K, V in the context of hourly load.
- Draw the network architecture.
- Report MAE, RMSE, and MAPE.
- Compare transformer and persistence.
- Compare transformer and LSTM.
- Discuss accuracy, training cost, and overfitting.

10. تقرير المخبر

يجب أن يتضمن تفسير Q وK وV، رسم البنية، مؤشرات الخطأ، مقارنة Transformer بخط الأساس وLSTM، ومناقشة الدقة والكلفة وفرط التخصيص.

References | المراجع

المراجع العامة المستخدمة في هذا المقرر

[1] A. Gabash, *Flexible Optimal Operations of Energy Supply Networks with Renewable Energy Generation and Battery Storage*. Saarbrücken, Germany: Südwestdeutscher Verlag für Hochschulschriften, 2014.

[2] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th Global ed. Harlow, U.K.: Pearson Education Limited, 2022.

[3] A. Gabash, “Energy Market Transition and Climate Change: A Review of TSOs-DSOs C+++ Framework from 1800 to Present,” *Energies*, vol. 16, no. 17, Art. no. 6139, 2023, doi: 10.3390/en16176139.

[4] A. Gabash, *AI Applications in Electrical Power Systems*, Version 1.0, 2026. [Online]. Available: https://aoussgabash.com

Publication Information | معلومات النشر

Document	Laboratory 10 · Load Forecasting with a Transformer
Course	AI Applications in Electrical Power Systems
Author	Dr.-Ing. Aouss Gabash
Version	1.0
Publication year	2026
Official website	https://aoussgabash.com
Citation style	IEEE

Recommended citation:
Gabash, A. (2026). Load Forecasting with a Transformer. AI Applications in Electrical Power Systems, Laboratory 10, Version 1.0. https://aoussgabash.com

المرجع المقترح:

أوس غباش، 2026، مخبر 10، تطبيقات الذكاء الاصطناعي في أنظمة الطاقة الكهربائية، الإصدار 1.0، https://aoussgabash.com

© 2026 Dr.-Ing. Aouss Gabash. Educational use with attribution to the author and official website.