

# Non-Intrusive Load Monitoring

Infering appliances power consumption patterns from power signals

Letteratura  
●○○○○○○○○

Implementazione  
○○○○○○○

Risultati  
○○○

Letteratura

# Introduzione

Il **Non-Intrusive Load Monitoring** è una tecnica di analisi dei segnali della rete elettrica al fine di una ricostruzione dei pattern di consumo degli apparati ad essa connessi.

Nell'articolo di riferimento <sup>1</sup> si studia l'implementazione di una Temporal Convolutional Network prendendo in esame due dataset: il *"UK Domestic Appliance-Level Electricity"* ed il *"REDD: A public data set for energy disaggregation research"*.

---

<sup>1</sup>Yang, Yandong, et al. "Semi-Supervised Multi-Label Deep Learning based Non-intrusive Load Monitoring in Smart Grids." IEEE Transactions on Industrial Informatics (2019).

# Dataset

## UK Domestic Appliance-Level Electricity

Contiene registrazioni dei consumi energetiche di cinque abitazioni, per ognuna delle quali è presente un record dell'edificio e record distinti dei singoli apparati presenti al loro interno.

Il periodo di campionamento è fissato a 6 secondi per il dataset, mentre per tre abitazioni su cinque (edifici 1, 2, 5) è inoltre presente una registrazione dei parametri di tensione e corrente, per l'intero edificio, campionati a 16 kHz.

# Dataset

## REDD: A public data set for energy disaggregation research

Contiene registrazioni dei consumi energetiche di sei abitazioni, per ognuna delle quali è presente un record dell'edificio e record distinti dei singoli apparati presenti al loro interno.

Il periodo di campionamento è fissato a 3 secondi per gli edifici, 1 secondo per gli apparati. Essendo uno dei primi dataset realizzati per il NILM, derivante da misurazioni aggregate e successivamente rifinite, presenta alcune inconsistenze quali la tipologia di potenza misurata e la disomogenea numerosità dei campioni.

Presente un set di registrazioni con campionamento a 16 kHz.

## Semi-Supervised Deep Learning

Dall'articolo<sup>2</sup>:

*“It has been demonstrated that deep learning can yield impressive results for many supervised learning tasks by leveraging large amounts of labeled observations. **However, collecting and annotating large datasets can be time consuming and expensive** [...] An attractive approach to address this challenge is semi-supervised deep semi-supervised multi-label learning framework.”*

---

<sup>2</sup>Yang, Yandong, et al. “Semi-Supervised Multi-Label Deep Learning based Non-intrusive Load Monitoring in Smart Grids.” IEEE Transactions on Industrial Informatics (2019).

# Modello proposto

## Temporal Convolutional Network

Rete neurale basata sul principio delle RNN, dove input e output possono non avere dimensioni predefinite e, generalmente, rappresentano sequenze o segnali nel tempo.

Questa variante è caratterizzata da:

- Stretta causalità delle convoluzioni, nessun *leakage* dal “futuro”
- Assenza di gating
- Memoria a lungo termine garantita da un variabile numero di layer *Residual* sovrapponibili

# Modello proposto

## Blocco residuale

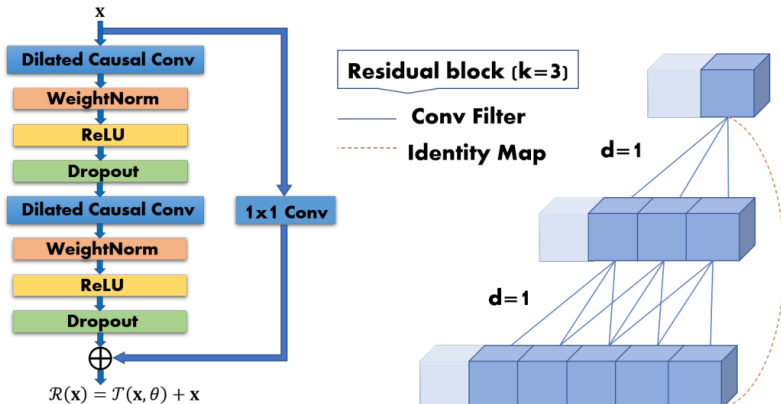


Figure 1: *Blocco residuale*



## Modello proposto

### Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks

*“We present weight normalization: a reparameterization of the weight vectors in a neural network that decouples the length of those weight vectors from their direction. By reparameterizing the weights in this way we improve the conditioning of the optimization problem and we speed up convergence of stochastic gradient descent. Our reparameterization is inspired by batch normalization but does not introduce any dependencies between the examples in a minibatch.”*<sup>3</sup>

---

<sup>3</sup>Salimans, Tim, and Durk P. Kingma. “Weight normalization: A simple reparameterization to accelerate training of deep neural networks.” Advances in neural information processing systems. 2016.

## Modello proposto

### Knowledge distillation

Architettura Student-Teacher:

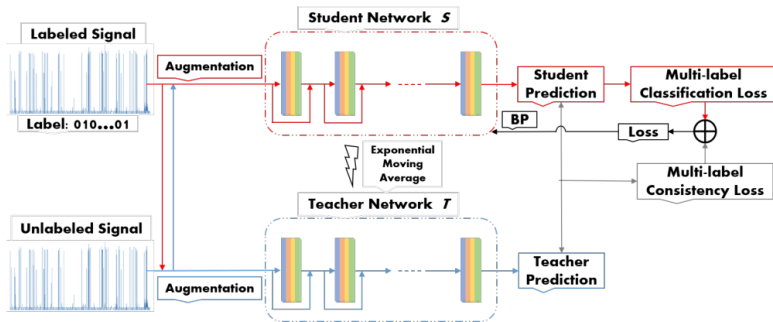


Figure 2: Architettura "Student-Teacher"

# Implementazione

# Dataset

## Importazione

- Edifici: *UK\_DALE*: 1,2; *REDD*: 1,3
- *Appliances*: ['kettle', 'microwave', 'dish washer', 'washing machine', 'fridge']
  - *mains* con periodo di campionamento di 30s per l'intero edificio
  - *average\_energy\_per\_period* con periodo pari a  $64 \cdot 30s$  per l'intero edificio
  - *power* con periodo di campionamento di 30s per le singole *appliance*
  - *when\_on* con periodo di campionamento di  $64 \cdot 30s$  per le singole *appliance*
- Preprocessing: MinMaxScaler e divisione in samples da 64 elementi, 32 minuti

# Temporal Convolutional Network

## Modello

### Elementi del blocco residuale

- Convoluzione 1D:
  - numero filtri = 64
  - dimensione kernel = 3
  - dilation rate =  $2^{(d-1)}$
  - padding "causal"
  - funzione di attivazione = 'relu'
- Dropout spaziale 1D, probabilità = 0.1
- Convoluzione 1D di raccordo

### Blocco residuale

- Convoluzione 1D (*normalizzato*)
- Dropout spaziale 1D
- Convoluzione 1D (*normalizzato*)
- Dropout spaziale 1D
- output sommato con la Convoluzione 1D di raccordo

## TCN

- Blocco Residuale,  $d = 1$
- Blocco Residuale,  $d = 2$
- Blocco Residuale,  $d = 3$
- Layer Dense, 5 neuroni, attivazione 'sigmoid'

# Temporal Convolutional Network

Reti

## Student:

- Modello TCN con normalizzazione dei pesi
- Ottimizzatore *Adam*, learning rate = 0.002

## Teacher:

- Modello TCN privo di normalizzazione
- Trasferimento dei pesi da rete student secondo una Exponential Moving Average

# Temporal Convolutional Network

## Loss

### Funzione di loss

Somma di due contributi:

1. Crossentropia binaria multi-label tra predictions della rete *student* e i labels (*supervised*)
2. Errore quadratico medio tra predictions della rete *student* e il riferimento *teacher* (*unsupervised*)

L'influenza della rete teacher è pesata da un coefficiente moltiplicativo che, nelle prime 80 epoche, raggiunge il valore di 50 con andamento sigmoidale per poi stabilizzarsi. Questo garantisce alla rete teacher sufficiente tempo per il training.

# Condizioni di training

## Metriche

- Test loss
- Hamming loss
- F1 macro, F1 micro
- Binary Accuracy

## Early stopping

Early stopping con patience e starting point variabili



## Condizioni di training

Eseguito sulla combinazione dei seguenti parametri:

### Training

- Epoche: 200
- Edifici:
  - singolo (*DALE 1 ; REDD 1*)
  - completo (*UKDALE 1,2*)
  - completo (*REDD 1,3*)
  - combinato (*UKDALE 1,2 + REDD 1,3*)
- Label drop ratio: 0.0, 0.3
- Rete Teacher: *presente, assente*
- Dimensione del minibatch: 256, 1024

### Testing

- Train-Test ratio: 0.2
- *Seen*
- *Unseen su DALE 2*
- *Unseen su REDD 3*

# Risultati

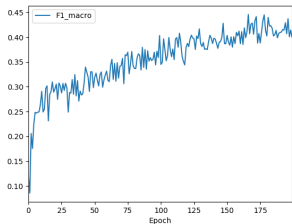
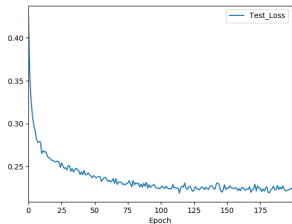
# Risultati

Migliori 20 per F1 macro

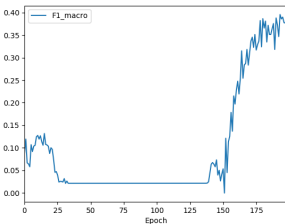
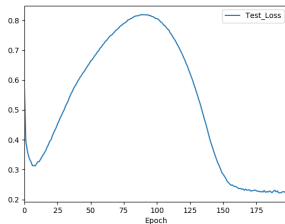
Parameters	Epoch	Test Loss	Test Accuracy	ANE	Micro	Macro
S:dh1 Drop:0.0 T:False BS:256 U:None	166	0.2235	0.8990	1.00	0.62	0.45
S:dh1 Drop:0.3 T:False BS:256 U:None	191	0.2316	0.8943	1.00	0.59	0.44
S:dh1 Drop:0.0 T:True BS:256 U:None	193	0.2227	0.8909	1.00	0.58	0.39
S:rh1 Drop:0.3 T:False BS:1024 U:[rh3]	197	0.1829	0.9539	0.80	0.87	0.35
S:dh1 Drop:0.3 T:True BS:256 U:None	191	0.2293	0.8878	1.00	0.57	0.34
S:rh1 Drop:0.0 T:False BS:1024 U:[rh3]	170	0.1847	0.9387	0.80	0.82	0.33
S:dh1,dh2 Drop:0.3 T:True BS:256 U:None	142	0.4460	0.8967	1.00	0.70	0.31
S:dh1,dh2 Drop:0.3 T:True BS:1024 U:[rh3]	22	0.4990	0.9672	1.00	0.91	0.30
S:rh1 Drop:0.0 T:False BS:256 U:[rh3]	176	0.2021	0.9532	0.80	0.88	0.28
S:rh1 Drop:0.0 T:True BS:256 U:None	24	0.5895	0.6328	0.80	0.50	0.27
S:rh1,rh3 Drop:0.3 T:False BS:256 U:None	157	0.3005	0.8922	0.80	0.74	0.27
S:dh1,rh1,dh2,rh3 Drop:0.3 T:True BS:1024 U:None	122	0.5813	0.5518	1.00	0.45	0.27
S:rh1,rh3 Drop:0.0 T:False BS:256 U:None	187	0.2882	0.8934	0.80	0.74	0.27
S:dh1,dh2 Drop:0.0 T:True BS:256 U:None	13	0.2306	0.9042	1.00	0.64	0.27
S:rh1 Drop:0.3 T:True BS:1024 U:[rh3]	141	0.6263	0.7615	0.80	0.59	0.26
S:dh1 Drop:0.3 T:True BS:1024 U:[rh3]	22	0.5195	0.7357	1.00	0.54	0.26
S:rh1 Drop:0.3 T:False BS:256 U:[rh3]	198	0.1987	0.9526	0.80	0.88	0.26
S:dh1,dh2 Drop:0.3 T:False BS:256 U:None	170	0.1983	0.8877	1.00	0.41	0.25
S:rh1 Drop:0.3 T:True BS:256 U:None	90	0.6379	0.6125	0.80	0.50	0.25
S:dh1 Drop:0.0 T:True BS:1024 U:[rh3]	20	0.4487	0.8135	1.00	0.59	0.25

## Influenza dei parametri

S:dh1 Drop:0.0 T:False BS:256 U:None



S:dh1 Drop:0.0 T:**True** BS:256 U:None



S:dh1 Drop:0.0 T:True BS:256 U:**rh3**

