Semi-Supervised Multi-Label Deep Learning Based Non-intrusive Load Monitoring in Smart Grids

Corso di Digital Adaptive Cricuits for Learning Systems
Università Politecnica delle Marche

Simone Crisologo

Contents

1 Introduzione

Il Non-Intrusive Load Monitoring è una tecnica di analisi dei pattern di assorbimento energetico di una rete elettrica, misurati a monte di essa, al fine di dedurre quali siano gli elettrodomestici e generici apparati in funzione e stimarne, per ognuno, il consumo. Quest'ultimo processo, in particolare, prende il nome di *disaggregazione*.

Oltre a rappresentare una interessante sfida tecnologica, è considerata, in ambito commerciale, un'alternativa economica all'uso di dispositivi di monitoraggio energetico per ogni singolo apparato connesso. Le rilevazioni effettuate sono misure di potenza attiva e reattiva, con frequenze di campionamento dipendenti dall'implementazione ma normalmente costituite da una frequenza bassa per la rilevazione dei consumi, con periodo nell'ordine dei secondi, e da una frequenza elevata per l'analisi dei transienti e la *failure detection*, entrambe accompagnate da *timestamp* opportuni.

Nell'articolo di riferimento [4] si esamina il ruolo degli strumenti forniti dalle tecniche di Deep Learning per l'analisi di tali segnali, con particolare focus sulle opportunità che queste offrono per il superamento di alcuni comuni ostacoli nel processamento dei dati, quali l'onerosità delle procedure di labeling nella realizzazione di nuovi dataset etichettati e la loro generale scarsa disponilibità. Viene illustrato come, in presenza di misurazioni prive o solo parzialmente fornite di label, la flessibilità e capacità di generalizzazione delle reti convoluzionali possa sopperire fornendo predizioni congrue e con una elevata affidabilità.

Il modello proposto, definito *Temporal Convolutional Network* e basato su tecniche convoluzionali monodimensionali, viene esaminato in diverse condizioni di operatività e viene valutato l'effetto dell'introduzione di una architettura "*Studet-Teacher*" nell'ottica della *knowledge distillation*. Il processo training prende in riferimento due principali dataset, il *"UK Domestic Appliance-Level Electricity"* ed il *"REDD: A public data set for energy disaggregation research"*, dai quali si esaminano quattro edifici ed un set di *"appliance"* predefinito.

2 Dataset

2.1 UK Domestic Appliance-Level Electricity

L'*energy disaggregation* è un'area di ricerca particolarmente attiva e che necessita la disponibilità di ingenti quantitativi di dati. Il dataset "UK Domestic Appliance-Level Electricity" si propone come primo dataset britannico di libero accesso contenente registrazioni ad alta risoluzione che arrivano a coprire un'estensione di 655 giorni.

I tempi di campionamento sono di 6 secondi per i cinque edifici presenti nel dataset, con l'aggiunta di un segnale campionato a 44 kHz, in downsampling a 16 kHz, registrato al contatore di tre di questi. Particolare attenzione è stata posta nella creazione e uso di un sistema di misurazione economico che fosse costituito da hardware e software liberi ed al quale è dedicata una sezione dell'articolo [1] pubblicato con il dataset.

Il dataset ha subito una revisione nel 2017 ed è stato ulteriormente arricchito. Le misurazioni sono disponibili sia per i singoli apparati collegati alla rete domestica, sia per l'intero edificio scelto, come valori di potenza attiva e apparente.

2.2 Reference Energy Disaggregation Data Set

Pubblicato dal Massachussets Institute of Technology nel 2011[2], viene considerato il primo dataset completo di libero accesso per lo studio delle tecniche di disaggregazione dei profili energetici. Come tale, a seguito della standardizzazione delle metodologie di analisi e, conseguentemente, delle tipologie di dati richiesti, ha subito un processo di revisione e consolidamento che ha ridotto il numero di edifici dai dieci presenti al momento della pubblicazione ai sei oggi disponibili. Per ognuno di questi è presente un record dell'energia misurata al contatore principale e record distinti dei singoli apparati connessi alla rete.

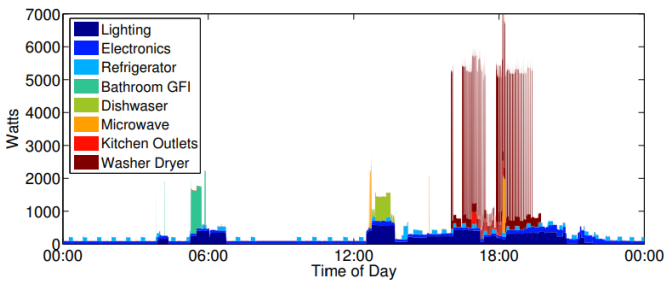


Figure 1: Esempio di consumo energetico giornaliero

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. Viene fornito un set secondario di registrazioni ad alta frequenza con campionamento a 16 kHz.

2.3 Semi-Supervised Deep Learning

Dall'articolo[4]:

> *"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 learning. In this section [...] the NILM problem is formulated in a semi-supervised multi-label learning framework."*

Una delle difficoltà tipiche nel *deep learning* è la creazione di dataset completi e correttamente etichettati. Non è infrequente dover revisionare manualmente dati generati tramite apparecchiature quali, nel caso qui trattato, contatori energetici, con l'obiettivo di consolidarli e di rimuovere eventuali discordanze tra *set* e *labels*. Questo compito risulta essere uno dei più onerosi nel momento in cui l'estensione del dataset risulta non trascurabile, portando a posticipare la pubblicazione di dati altrimenti pronti. In generale, la disponibilità di dati correttamente etichettati e di libero accesso è fattore limitante nello sviluppo e studio di nuovi modelli nell'ambito dell'analisi dei dati.

Una soluzione di recente adottata e qui ripresa vede l'uso delle tecniche di deep learning non solamente per la predizione, al fine ultimo della *fault detection* o della stima dei consumi, nel caso di dataset fruibili e completi, ma anche come riferimento attendibile in presenza di dati grezzi non precedentemente valutati dall'operatore umano. In questo elaborato il focus principale è la costruzione di un modello di rete convoluzionale robusto che sia in grado di evolvere anche in condizioni di *Semi-Supervised* learning, ovvero in assenza di label in input. A tale scopo viene presentata una variante delle tecniche di *knowledge distillation* che prevede la presenza di due reti parallele, nel ruolo di elemento *Student* ed elemento *Teacher*, le quali concorrono alla corretta predizione sul dato fornito.

3 Temporal Convolutional Network

Storicamente tra le più diffuse ed efficaci reti di processamento di segnali, le Reti Neurali Ricorrenti sono state per lungo tempo considerate il metodo di riferimento per task quali il riconoscimento di sequenze continue non precedentemente segmentate. La loro struttura è una alterazione delle classiche reti *feedforward* e presenta un elemento di "memoria" delle iterazioni passate che contribuisce alla generazione dell'output.

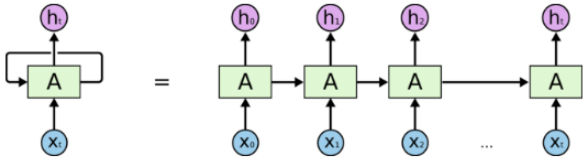


Figure 2: Unità ricorrente interpretata come sequenza

Loro evoluzione, le reti definite "Long Short-Term Memory" hanno successivamente risolto alcuni dei limiti di una architettura così semplice come il problema dell'*evanescenza o esplosione del gradiente* e la difficoltà nel processare sequenze particolarmente estese ritenendo contenuto informativo appartenente a campioni passati distanti.

Per fare ciò sono stati introdotti meccanismi articolati di *gating* (Figura 3) che regolano o precludono totalmente la propagazione di contenuto dall'elemento di memoria variandone l'influenza che questa ha sul campione correntemente analizzato. I meccanismi con cui questi operano si basano sull'output di una funzione sigmoidale per la determinazione della presenza od assenza dell'effetto del contributo di memoria e sull'output di una funzione *tanh* per pesare in maniera opportuna tale effetto.

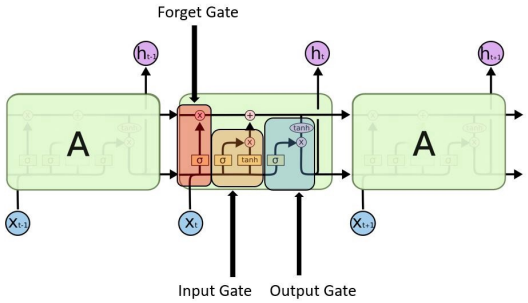


Figure 3: Meccanismi di gating di una LSTM

Entrambe queste tipologie soffrono notoriamente di una complessità implementativa che si traduce in elevati tempi di elaborazione e necessità di dataset estesi. Una adeguata capacità di memorizzazione richiede spesso, inoltre, l'estensione delle dimensioni della rete, gravando ulteriormente sulle capacità di *processing* dell'elaboratore adoperato.

Il modello qui proposto abbandona il meccanismo di ricorrenza delle RNN ereditando la struttura più comune delle Reti Neurali Convoluzionali ed adattandone la dimensionalità. Questa scelta ha il duplice effetto di semplificare le operazioni aritmetiche introdotte, eliminando l'uso di funzioni non lineari, ed uniformare quindi il processamento dei dati, rendendolo particolarmente adatto ad implementazioni ottimizzate su unità computazionali grafiche.

3.1 Blocco Residuale

L'unità logica implementata è definita *Blocco Residuale*. Essa è stata introdotta come soluzione agli effetti di saturazione dell'accuratezza ed alla generica difficoltà delle reti costituite da numerosi layer di preservare l'abilità di riprodurre comportamenti elementari quali la funzione identità. Queste problematiche, note nell'ambito come "*degradation problem*", vengono eliminate tramite l'adozione di blocchi di profondità ridotta che presentino, parallelamente al ramo di propagazione principale, delle *skip connection*, ovvero delle linee di trasmissione che riportino il segnale di input sull'output mantenendolo inalterato.

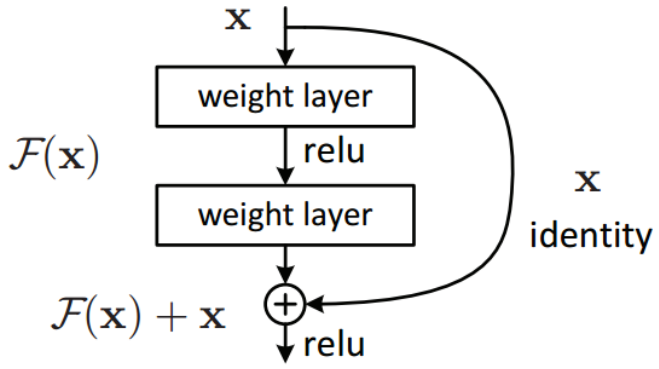


Figure 4: Blocco residuale

Il nome deriva dalla particolare formulazione che assume il problema dal punto di vista matematico, dove l'obiettivo della rete non è più l'approssimare l'uscita richiesta a partire dal solo campione in ingresso ma riporodurre correttamente l'alterazione che l'input subisce, modellata come differneza dei due segnali e chiamata "residuo". Definita $H(x)$ la funzione che si vuole approssimare, data

x sequenza in ingresso, si può esprimere il residuo come:

$$Res(x) = (Outputs - Inputs) = H(x) - x$$
$$H(x) = Res(x) + x$$

Priva di feedback, la capacità di memorizzazione della rete viene data dalla numerosità dei layer introdotti, dall'abilità della rete nell'astrazione del contenuto e da un fattore di decimazione delle interconnessioni all'aumentare della profondità del modello, risultante in dilatazione temporale. Viene inoltre garantita totale causalità della rete, impedendo il propagarsi di contributi di segnale "futuro" agli elementi che li precedono. Nella particolare variante qui introdotta il blocco residuale presenta una duplice convoluzione monodimensionale ad attivazione lineare rettificata (ReLU) con l'introduzione di un fattore di *dropout* delle connessioni ed un processo di normalizzazione dei parametri ottimizzati.

3.2 Weight Normalization

Ulteriore elemento di ottimizzazione del training è costituito da un processo di normalizzazione dei pesi del modello, ispirato alla *batch normalization*, così descritto dall'articolo[3] che lo introduce:

> *"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."*

Questo processo consente, quindi, di scindere gli effetti del modulo e della direzione dei vettori rappresentanti i pesi delle interconnessioni del modello durante la fase di ottimizzazione, portando a una maggior rapidità di convergenza e, tipicamente, performance del modello più elevate.

3.3 Knowledge distillation

Al fine di ampliare la capacità di memorizzazione ed aumentare la robustezza a condizioni di training non ideali come la variante non supervisionata, viene introdotto un meccanismo di *knowledge distillation* rappresentato dall'uso concomitante di due modelli di Temporal Convolutional Network, la rete *Student* e la rete *Teacher*, nelle quali il processo di "learning" consiste nell'ottimizzazione classica dei parametri appartenenti al modello *Student* e nel successivo trasferimento di questi al modello *Teacher*.

Il processo di trasferimento è inoltre soggetto a filtraggio tramite una *Exponential Moving Average*, dotando di fatto la TCN *Teacher* di resilienza alle variazioni repentine di segnale (relativamente allo span temporale coperto dall'estensione del sample in ingresso).

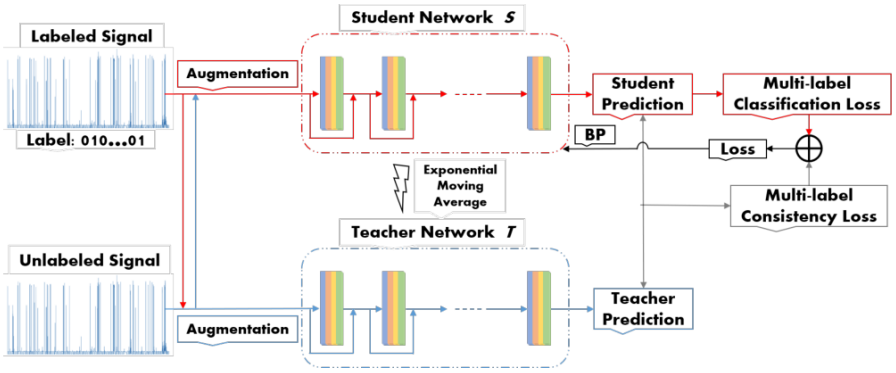


Figure 5: Architettura "Student-Teacher"

Il processo di training avviene fornendo dati etichettati alla rete *Student*, la quale effettua predizioni sulla base della propria esperienza valutate con una cross-entropia binaria, e dati di riferimento non etichettati alla rete *Teacher*, che restituisce una predizione costituente ulteriore fattore di valutazione nella determinazione dell'errore su quanto stimato dalla rete *Student* ed elemnto chiave nella predizione nel caso di dataset nuovi e privi di etichette. Questo elemento correttivo viene pesato da un fattore variabile secondo legge sigmoidale lungo l'arco delle prime 80 epoche, affinché il modello possa prima acquisire un'adeguata esperienza su dati correttamente etichettati.

4 Implementazione

4.1 Importazione dei dati

Dai dataset sono stati scelti quattro edifici, due per ogni set, e cinque elettrodomestici tipici quali *kettle*, *microwave*, *dish washer*,*washing machine*, *fridge*. Di questi, 'kettle' non risulta presente nei record del REDD ed è stato semplicemente ignorato dove assente.

Il periodo di campionamento scelto è di 30 secondi con interpolazione lineare dei dati per poter coprire uno span temporale più ampio senza un aumento considerevole della complessità della rete, con una numerosità del *sample* pari a 64 elementi per un totale di 32 minuti. I dati presi in considerazione sono:

- Edifici: *UK_DALE*: 1,2; *REDD*: 1,3
- *Appliances*: kettle, microwave, dish washer,washing machine, fridge
  - Registrazione al contatore principale con periodo di campionamento di 30s
  - Label con periodo di campionamento di $64 \cdot 30$s per le singole *appliance*
  - Energia assorbita, istantanea e media, per le singole *appliance*

Come precondizionamento dei dati è stata eseguita una normalizzazione, condotta separatamente su ogni edificio, secondo la metodologia *MinMax*. I sample così ottenuti vengono poi introdotti nel modello, in fase di training, tramite mini-batching variable.

4.2 Temporal Convolutional Network

Come anticipato, il modello include blocchi residuali composti da due convoluzioni monodimensionali ad attivazione lineare rettificata seguiti da una decimazione probabilistica delle connessioni. Questa, in particolare, differisce dall'implementazione classica poiché opera una recisione dei collegamenti tra layer a livello dell'intero sample non solo per i singoli elementi del campione. Tale tecnica di dropout viene chiamata "spaziale". Come mostrato dall'illustrazione, il blocco residuale possiede le seguenti caratteristiche:

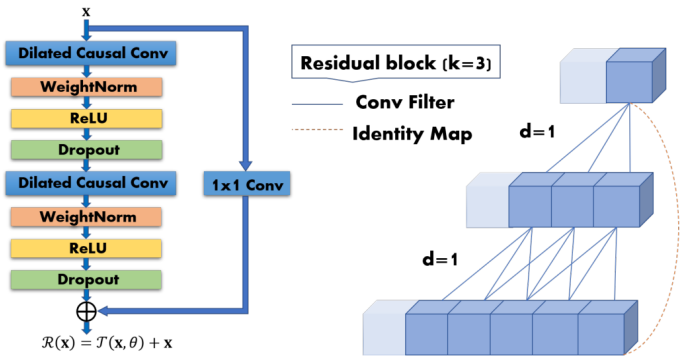


Figure 6: Blocco residuale TCN

Blocco Residuale	
Parametro	Dato
Numerosità filtri	64
Dimensione kernel	3
Dilation rate	$2^{(d-1)}$
Padding della convoluzione	'causal'
Funzione di attivazione	'ReLu'
Spatial Dropout	0.1

La numerosità dei filtri scelta permette di coprire un'estensione temporale di 32 minuti di registrazione in ogni campione, mentre il *dilation rate* è funzione della profondità della rete: il modello di Temporal Convolutional Network qui presentato prevede la presenza di 3 blocchi residuali, per un *dilation rate* variabile nell'insieme $1, 2, 4$. Il modello termina con un layer di tipo *Fully Connected* costituito da 5 neuroni ad attivazione sigmoidale.

TCN Model		
Layer	Dilation Rate	Shape
Residual block	$r = 1$	$(n, 1, 64)$
Residual block	$r = 2$	$(n, 1, 64)$
Residual block	$r = 4$	$(n, 1, 64)$
Layer Dense	–	$(n, 1, 5)$

Nel caso della rete *Student* è stato inserito un wrapper di normalizzazione per le convoluzioni, non utilizzato al contrario nella rete *Teacher* perché, a causa di modalità implementative, impediva il corretto svolgersi della fase di training in quanto, tecnicamente, la rete coinvolta non subiva processi di ottimizzazione tramite calcolo del gradiente ma solo un trasferimento dei parametri.

L'ottimizzatore scelto per la rete *Student* è stato quindi di tipo "Adam" con un fattore di *learning rate* di 0.002.

4.3 Funzione di loss

La funzione di *loss* ideata è rappresentativa dei contributi di training di tipo *supervised* e *unsupervised* mediante due contributi principali L_l, L_u.

Per la variante *supervised*, al fine di eseguire una *detection* di molteplici apparati, viene adoperata una *Binary Crossentropy* media, calcolata separatamente su ogni *appliance*. Esprimendo l'output della rete *Student* come:

$$\hat{y} = g_\Theta(x) \tag{1}$$

con g_Θ rete di parametri Θ, x l'input proposto, la funzione di *loss* applicata segue la definizione:

$$L_l(y, \hat{y}) = \frac{1}{|A|} \sum_i y_i log(\sigma(\hat{y}_i)) + (1 - y_i) log(1 - \sigma(\hat{y}_i)) \tag{2}$$

dove $\sigma(\hat{y}_i) = \frac{1}{1+e^{-\hat{y}_i}}$ è la funzione sigmoidale la cui uscita è limitata nel range $[0, 1]$.

Nella variante *unsupervised* la comparazione degli output delle reti *Student* e *Teacher* avviene tramite il calcolo del *Mean Square Error* sulle predizioni ottenute.

Definito $\hat{y}$ l'output della rete *Student* e $\tilde{y}$ quello della rete teacher, si ha:

$$L_u(\hat{y}, \tilde{y}) = \frac{1}{|A|} \sum_i (\sigma(\hat{y}_i) - \sigma(\tilde{y}_i))^2 \tag{3}$$

Entrambe i valori di loss così ottenuti contribuiscono al calcolo dell'errore ottenuto dal modello adoperante parametri Θ per la rete *Student* e una *Exponential Moving Average* degli stessi per la rete *Teacher*, aggiornata al termine di ogni epoca:

$$\Theta'_t = \alpha \Theta'_{t-1} + (1 - \alpha)\Theta_t \tag{4}$$

In ultimo, la loss viene espressa come somma pesata di L_u, L_l:

$$L = L_l + \omega(\tau) * L_u \tag{5}$$

dove il coefficiente moltiplicativo $\omega = e^{-5(1-x)^2}, x \in [0, 1]$ funge da funzione di *ramp-up* nel corso delle epoche per regolare l'influenza delle predizioni *unsupervised* al fine di evitare perturbazioni del sistema nelle prime fasi del training che possano comprometterne la convergenza.

4.4 Metriche

Le metriche adoperate sono state scelte affinché rispecchiassero adeguatamente la tipologia di problematica trattata e si adeguassero alle caratteristiche del dataset, quali ad esempio l'elevata presenza di *veri positivi* sulla totalità dei sample trattati. Al termine di ogni epoca viene quindi espressa la *Hamming Loss* rilevata, i principali *F1 score* nelle varianti *macro* e *micro* ed una misura della *loss* effettuata sulla porzione di dati dedicata alla fase di test. La *Hamming Loss* viene espressa come:

$$hloss(g) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|A|}|g(x_i)\Delta y_i| \tag{6}$$

ed indica la frazione di label correttamente predetti dalla rete.
Le medie *macro, micro* sono metriche derivate dal più generale $F1_{score}$:

$$F1(TP, FP, FN) = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{7}$$

$$F1_{micro} = F1(\sum_{i=1}^{|A|} TP_i, \sum_{i=1}^{|A|} FP_i, \sum_{i=1}^{|A|} FN_i) \tag{8}$$

$$F1_{macro} = \frac{1}{|A|}\sum_{i=1}^{|A|} F1(TP_i, FP_i, FN_i) \tag{9}$$

L'articolo di riferimento aggiunge, inoltre, una misura dell'errore sulla predizione interpretata come differenza tra l'energia effettiva assorbita dal singolo apparato ed energia assorbita stimata dalla rete, con il nome di *Average Normalized Error*:

$$ANE = \frac{|\sum_{i=1}^{N} P_i - \sum_{i=1}^{N} \hat{P}_i|}{\sum_{i=1}^{N} P_i} \tag{10}$$

4.5 Condizioni di training

Il processo di training è stato condotto su una combinazione di parametri variabili allo scopo di testare il modello in condizioni differenti. Una panoramica di questi viene presentata nella tabella sottostante:

Parametro	**Set**
Epoche	200
Dataset	• singolo edificio *DALE:1, REDD:1* • set completi *DALE, REDD* • set combinati *DALE + REDD*
Condizione di test	• modalità *seen* • modalità *unseen* su *DALE:2* • modalità *unseen* su *REDD:3*
Train-Test ratio	0.2
Rete *Teacher*	*Presente, Assente*
Label Drop Ratio	0.0, 0.3
Dimensione del minibatch	256, 1024

Non è stato adoperato alcun meccanismo di *early stopping*, benché disponibile ed utilizzato dall'articolo, al fine di ottenere dati quanto più facilmente comparabili. Questo è risultato occasionalmente in peggioramenti delle performance delle reti una volta superato il numero di epoche ideale per la particolare combinazione di parametri scelta, di cui bisognerà quindi tener conto nell'osservare i dati riportati.

5 Risultati

I risultati qui riportati (Tabella 1), rappresentanti le migliori 20 combinazioni ordinate per punteggio ottenuto nella metrica F1 macro, mostrano performance non in linea con quanto riportato dall'articolo imputabili ad imprecisioni tecniche nell'implementazione.
È possibile evidenziare, però, alcuni comportamenti del modello adoperato, come performance tipicamente superiori per dataset più piccoli (dovute alla minor varietà dei segnali presi in esame) e complessivo degrado delle stesse per condizioni di training con set *unseen*.

Dai grafici che seguono (Figure 7,8,9), scelti tra le varianti del training

Table 1: Primi 20 risultati 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

riguardanti il medesimo set (*dh1*) con variazioni su elementi chiave del modello, si possono evidenziare ulteriori particolarità, quali l'effetto dell'inerzia introdotta nella fase di *learning* dalla presenza del modello *Teacher*, visibile nella seconda serie di immagini, o, come anticipato, le performance inferiori ottenute nel caso di training con set *unseen*.

6 Conclusioni

In questo elaborato è stato presentato un modello di disaggregazione e classificazione di *appliance* domestiche basato su tecniche di Deep Learning versato alla metodologia di training *Semi-supervised*. Tramite una struttura simmetrica *Student-Teacher*, costruita sul principio della *Knowledge Distillation*, ne è stata

illustrata l'efficacia per i due dataset di riferimento *"UK Domestic Appliance-Level Energy"* e *"Reference Energy Disaggregation Data Set"* in condizioni di operatività non ideali quali la carenza di dati pienamente etichettati. Le performance ottenute nelle prove svolte restituiscono un comportamento della Temporal Convolutional Network presentata non in linea con quanto dimostrato nell'articolo[4] preso in esame ed indicano una necessità di revisione del metodo implementativo. Per quanto illustrato dallo studio di riferimento, la Temporal Convolutional Network evidenzia la spiccata abilità di astrazione dei modelli convoluzionali, paragonata a tecniche parallele di *Supervised Learning*, con una particolare versatilità d'impiego in presenza di dati scarsi, incompleti o non precedentemente valutati.

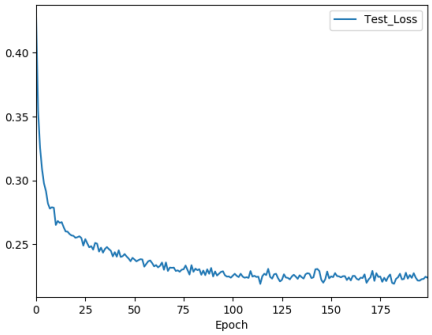
References

[1] Jack Kelly and William Knottenbelt. "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes". In: *Scientific data* 2.1 (2015), pp. 1–14.

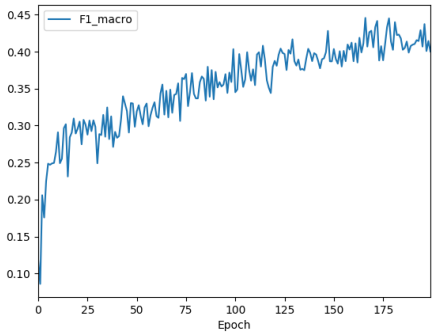
[2] J Zico Kolter and Matthew J Johnson. "REDD: A public data set for energy disaggregation research". In: *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*. Vol. 25. Citeseer. 2011, pp. 59–62.

[3] Tim Salimans and Durk P Kingma. "Weight normalization: A simple reparameterization to accelerate training of deep neural networks". In: *Advances in neural information processing systems*. 2016, pp. 901–909.

[4] Yandong Yang et al. "Semi-Supervised Multi-Label Deep Learning based Non-intrusive Load Monitoring in Smart Grids". In: *IEEE Transactions on Industrial Informatics* (2019).

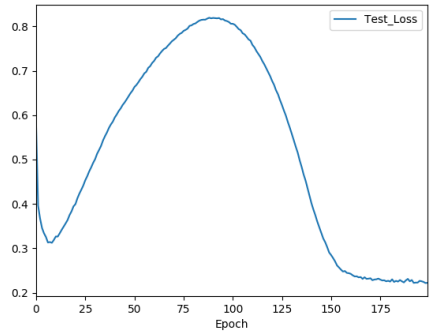


(a) Test Loss

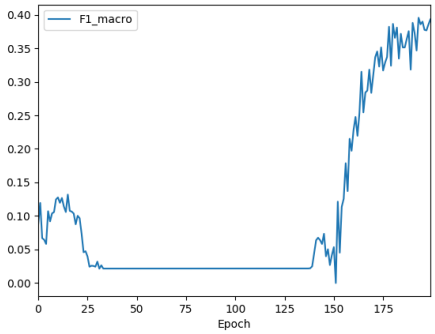


(b) F1 macro

Figure 7: S:dh1 Drop:0.0 T:False BS:256 U:None

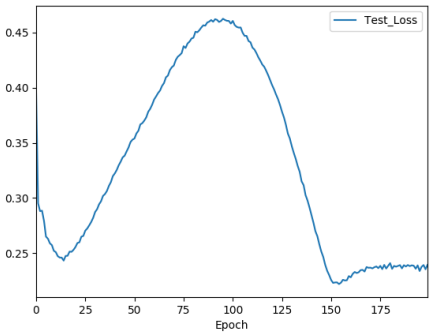


(a) Test Loss

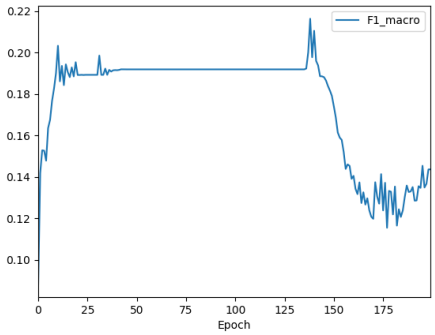


(b) F1 macro

Figure 8: S:dh1 Drop:0.0 T:**True** BS:256 U:None



(a) Test Loss



(b) F1 macro

Figure 9: S:dh1 Drop:0.0 T:True BS:256 U:**rh3**