

العدد السابعون / ديسمبر / 2023

تطوير نظام التعرف على الارقام المكتوبة بخط اليد باستخدام  
تقنيات الشبكة العصبية ANN و CNN

د. محمد سعد صالح

قسم الحاسوب - كلية العلوم  
جامعة عمر المختار

[mohammed.saad@omu.edu.ly](mailto:mohammed.saad@omu.edu.ly)

سامية موسى الثمن

قسم علم الحاسوب - كلية تقنية المعلومات - فرع المرج  
جامعة بنغازي

[samiacomp.sc@gmail.com](mailto:samiacomp.sc@gmail.com)



## تطوير نظام التعرف على الأرقام المكتوبة بخط اليد باستخدام تقنيات الشبكة العصبية ANN و CNN

### الملخص

أصبحت مشكلة التعرف على الأرقام المكتوبة بخط اليد واحدة من أهم مشاكل التعلم الآلي وتطبيقات الرؤية الحاسوبية، وقد تم استخدام العديد من تقنيات التعلم الآلي لحل مشكلة التعرف على الأرقام المكتوبة بخط اليد. تم في هذا الورقة تطوير نظام يقوم بتمييز ارقام المكتوبة بخط اليد باستخدام الخوارزميتين من خوارزميات التعلم الآلي وهي Artificial Neural Network (ANN) و Convolutional Neural Network (CNN)، وتم استخدام قاعدة بيانات خاصة بالأرقام المكتوبة بخط اليد وهي Modified National Institute of Standards and Technology database (MNIST). تم النتائج مقارنة النتائج بناء على الدقة والأداء مع طبقات Batch Normalization و Dropout. و نتائج كانت جيدة فقد تحصلت CNN على نسبة مع استخدام BN+PO 99% مقارنة مع ANN في تمييز الأرقام المكتوبة بخط اليد.

الكلمات الدالة : كتابة خط اليد، تعليم الآلة ، الشبكات العصبية

### Developing a system to predict handwriting numbers using ANN and CNN techniques

Samia Musa Althemen<sup>1</sup> Mohammed Saad Saleh<sup>2</sup>

#### Abstract

The problem of recognizing handwritten numbers has become one of the most important problems in machine learning and computer vision applications. Many machine learning techniques have been used to solve the problem of recognizing handwritten numbers. In this paper, a system was developed that distinguishes handwritten numbers using two machine learning algorithms, which are Artificial Neural networks (ANN) and Convolutional Neural networks (CNN). A special database for handwritten numbers, the Modified National Institute of Standards and Technology, was used. Database (MNIST). The results were compared based on accuracy and performance with the Batch Normalization and Dropout layers. The results were good. The CNN obtained a score of 99% with the use of BN+PO compared to the ANN in distinguishing handwritten numbers.

**Keywords :** Handwriting, Machine Learning, Artificial Neural networks

إن حاسة النظر هبة وهبنا الله إياها، بحيث يمكننا رؤية الأشياء المحيطة من حولنا والتعرف عليها بسهولة، ولكن حاسة النظر تعتبر عملية معقدة، حيث يتم إرسال إشارات إلى الدماغ والقشرة البصرية بالتحديد، حتى يتسنى لنا فهم ومعرفة ما رأيناه. مع التطور التقني الذي نشهده أهتم العلماء الباحثين والمطورين بدمج التكنولوجيا بعالمنا الواقعي، واحد من هذه المجالات هو الرؤية الحاسوبية Computer Vision، خصوصا مع انتشار العديد من الصور والفيديو والكاميرا على الانترنت، وتتفوق الحواسيب على الإنسان في الدقة العالية التي تتميز بها في التقاط الصور ومعرفة الألوان، إلا أن الإنسان لديه القدرة على فهم ومعرفة ما يحيط به أو ما يراه، بينما الحاسوب يرى الصور عبارة عن Pixels وقيم عددية تحدد الألوان فقط.

هذا الأمر أصبح يثير اهتمام العلماء، وهو جعل الحواسيب تفهم العالم الحقيقي المحيط بنا، وقد ظهرت عدة أبحاث وتم تطوير عدة نماذج استطاعت أن تتعرف وتصنف الأشياء. واحدة من هذه المجالات التي يندرج تحت الرؤية الحاسوبية وهو نظام يعمل على التصنيف الاشياء مثل التعرف على الوجه، التعرف على الكلام أو التعرف على خط اليد. التعرف على خط اليد هو نوع من التعرف الضوئي على الحروف Optical Character Recognition ويقصد به قدرة الألة على التعرف وتفسير النصوص المكتوبة بخط اليد، ويمكن أن تتم عملية التعرف بالاتصال بالإنترنت أو من دون انترنت. وفي هذا البحث تم التعرف على الأرقام المكتوبة بخط اليد باستخدام تقنية CNN و ANN و قاعدة البيانات MNIST. التعرف على الأرقام المكتوبة بخط اليد Handwritten Digit Recognition يتم عن طريق إدخال صورة مكونة من الرقم المراد التعرف عليه ثم يتم إخراج الرقم المفترض التعرف عليه. ساعدت هذه التقنية في حل العديد من المشاكل مثل التعرف على أرقام الشيكات المصرفية [1].

من الخوارزميات التي يمكن استخدامها وقد أظهرت مدى دقتها وفعاليتها في التعرف هي الشبكات العصبية الاصطناعية ANN وهي تقنية تحاكي دماغ الانسان . تقليد للشبكة العصبية البيولوجية ولكن بمنظور آخر، بمعنى تشابه في طريقة الأداء ولكن بشكل مختلف و الخوارزمية المستخدمة الثانية تسمى الشبكات العصبية الالتفافية CNN وهذه الشبكات نوع من أنواع الشبكات العميقة التي أثبتت فعاليتها وأدائها خصوصا في أحما مختصة بمجال التصنيف الصور هذا الأمر جعل منها أكثر استخداما.

مع التطور السريع الذي نشهده، والتوجه نحو العالم الرقمي. نلاحظ الاحتياج أكثر لتسهيل وتسريع العمل من ادخال وكشف وارسال سواء في المؤسسات الحكومية أو المدارس، أو المصارف. يسعى دائما الى رقمته المهام وأرشفتها بسهولة الوصول إليها، وهذا الأمر قد يكون متعبا ويحوي الكثير من الأخطاء بسبب الإدخال اليدوي لقاعدة البيانات. والهدف من هذا الورقة هو إنشاء نظام يستخدم الشبكة عصبية اصطناعية ANN وشبكة عصبية التفاضلية CNN باستخدام لغة Python مع استخدام قاعدة البيانات MNIST . وكذلك استخدام طبقات Batch Normalization و Dropout،

## العدد السابعون / ديسمبر / 2023

لتجنب مشكلة . هناك العديد من التقنيات لحل مشكلة Overfitting ،ومن هذه التقنيات هي ال Batch Normalization وهي واحدة من التقنيات التي تستخدم بشكل شائع في التعلم العميق ومن ميزاتنا أنها تحسن من سرعة الشبكات العصبية وتوفر التنظيم Regularization كما تمنع من حدوث Overfitting [2] . بعد تطبيق هذه المراحل نقوم مقارنة النتائج للتوصل الى افضل أداء لشبكة العصبية في التعرف على الأرقام المكتوبة بخط اليد.

### 2- الدراسات السابقة

#### 1-2 الدراسات السابقة للشبكة العصبية الاصطناعية Artificial Neural Network

زاد اهتمام الباحثين والأكاديميين في السنوات الأخيرة في العمل على زيادة كفاءة التعرف على الأرقام المكتوبة بخط اليد وتحسين دقته. يحتاج التعرف على الأرقام المكتوبة بخط اليد الى مجموعة بيانات كبيرة نسبيا وكذلك وقت طويل في التدريب لتحسين الدقة وتقليل نسبة الخطأ [1]. تم بناء نموذج لشبكة عصبية اصطناعية لتعمل بكفاءة وسرعة. وتعمل بطريقة التدريب المتوازية باستخدام وحدة معالجة الرسومات GPU للتقليل من وقت التدريب. وفي هذه الدراسة تم مقارنة وحدة معالجة الرسومات GPU مع النهج الحالي المعتمد على وحدة المعالجة المركزية CPU وتبين من المقارنة كفاءة وحدة معالجة الرسومات حيث وصلت لدقة في تصنيف الارقام بنسب بين 97-100 بالمئة لكل رقم بينما منهجية التي تعتمد على CPU كانت نسبها تصل بين 87-100 بالمئة. وفي الاختبار تم استخدام قاعدة البيانات MNIST لتعرف على الارقام من 0 إلى 9، باستخدام [2].

في هذه الورقة تم العمل على دراسة استقصائية Survey حول بعض المنهجيات الشبكات العصبية للتعرف على الأرقام المكتوبة بخط اليد. تم فيها مقارنة بين ثلاث خوارزميات ومن بينها خوارزمية الانتشار الخلفي التي قدمت أداء أفضل من السرعة والفعالية مقارنة بالباقي [3].

التعرف على الأرقام والحروف هي مهارة مطلوبة في صناعة التطبيقات الحالية مثل معالجة الاشارات، التعرف على العملات والتعرف على رقم المنزل. تركز الورقة على أداء الشبكة وفقا للمعالجة المسبقة للبيانات مع التدرج الرمادي بالإضافة إلى استخراج الميزات تم استخدام ANN للتعرف على الأرقام باستخدام Octave والتنظيم لتجنب مشكلة Overfitting. وقد تبين أن الشبكة تنفذ في وقت أقل مع دقة بنسبة 97.5% [4].

التعرف على الخط هو قدرة الحاسوب على استقبال وتفسير مدخلات مكتوبة بخط اليد من مصادر معينة مثل المستندات الورقية، الصور وشاشات اللمس . تمت المقارنة بين ثلاث خوارزميات وهي (SVM) و (KNN) و (ANN) وقد ركزت الدراسة في المقارنة على الدقة فيها تبين أن أفضل دقة تم الحصول عليها هي 99.26 بالمئة للخوارزمية KNN وتبين كذلك أن أداء كل من KNN و SVM أدائهما أفضل بكثير من MLP [5].

## العدد السابعون / ديسمبر / 2023

[6] دراسة مقارنة لثلاثة خوارزميات وهي (Naïve Bayes) و (MLP) و (K\_star) وقد تمت المقارنة بناء على دقة الأداء بحيث تركز الدراسة على أي الخوارزميات تعطي دقة أفضل مقابل عدد ميزات أقل Features. وقد أعطت خوارزمية K\_star دقة أعلى بنسبة وهي 82%.

### 2.2- الدراسات السابقة للشبكة الالتفافية Convolutional Neural Network

في السنوات الأخيرة الماضية أظهرت أساليب التعلم العميق، وبالأخص الشبكات العصبية تلافيفي CNNs أظهرت دقة ممتازة في الكثير من مشاكل تصنيف الأنماط Pattern Classification. تتضمن دراسة حالة للشبكات الالتفافية بميكليات مختلفة وقد ركزت هذه الدراسة على إعطاء لمحة عن المعالجة المسبقة للبيانات Pre-Processing و زيادة حجم بيانات التدريب Augmentation Data وبينت أهمية المعالجة المسبقة وكيف أن لها تأثير على مدى أداء الشبكة وكيف أن زيادة حجم البيانات يساعد على تكرار البيانات على الشبكة. مما يساعدها على التعلم. وتم المقارنة بين أنواع مختلفة من الشبكة الالتفافية. وأظهرت النتائج أداء جيد يصل الى 99% لكل شبكة على اختلاف هيكلية كل واحدة منها [7]. لقد أخذ التعلم العميق منحى وتغيرا جذريا وبذلك أصبح أكثر في حل المشاكل المعقدة. [8] وفي هذه الدراسة تم مقارنة ثلاث أنواع من الشبكات من نوع الشبكة العميقة وتركز على تحديد أفضل خوارزمية بناء على عدة عوامل من أهمها الدقة والأداء وكذلك آلية التنفيذ. تم استخدام مجموعة بيانات لتقييم الخوارزميات، من خلال النتائج تبين (DNN) أفضل خوارزمية، من حيث الدقة والأداء بينما الخوارزميات الأخرى (CNN) و (DBN) متشابهة ومتقاربة في الدقة. [9] تم تحسين بنية الـ (CNN) مع ضبط دقيق للمعلمات ومع استخدام مجموعة البيانات (MNIST) وتم تركيز العمل على تجنب المعالجة المسبقة للبيانات مع زيادة في عدد الطبقات والمقارنة بينها. وقد تبين من النتائج أن الشبكة ذات الأربع طبقات أدت إلى دقة أفضل وتصلت إلى 99.76 بالمئة. وتبين أن الضبط الدقيق للمعلمات يساعد كثيرا في تحسن أداء الشبكة بالإضافة لخوارزمية التحسين (Adam) التي حققت الشبكة بجانبه دقة لـ 99.89%.

تم في هذه الورقة تقديم تطبيق للتعرف على الأرقام المكتوبة بخط اليد، تم تطوير نموذج CNN يمكنه التعرف على الأرقام من الصور بحيث يمكنه التنبؤ بدقة عالية. تم التوصل لدقة 99.15% ويمكن لهذا النموذج من تحويل الأرقام الى صيغة رقمية [10]. [11].

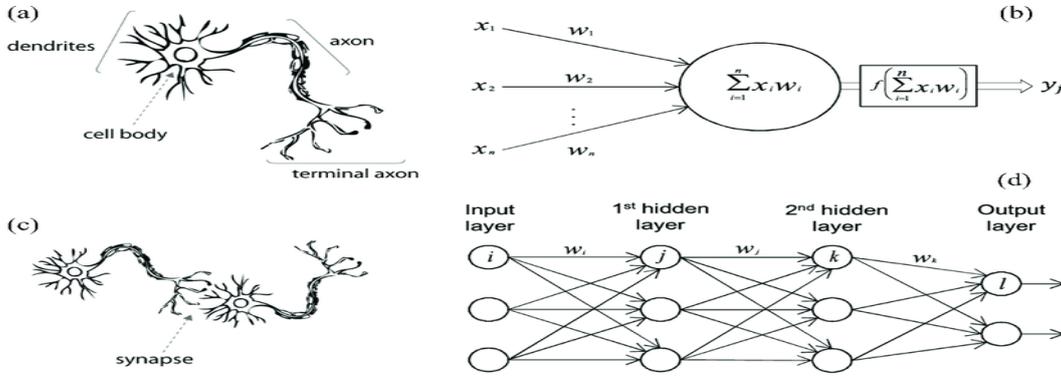
في هذه الورقة تم تصميم نظام التعرف على المستندات المكتوبة بخط اليد باستخدام تقنية الشبكة العصبية التلافيفية CNN. وفي هذه الدراسة تم اختبار البيانات المكتوبة بخط اليد باللغة الإنجليزية. وتم تدريب النظام المقترح لمجموعة كبيرة من عينات البيانات واختبارها على عينات الصور للمستندات المكتوبة بخط اليد. النتائج تم التعرف على معظم الحروف بنسبة 86%. ومع زياد عدد البيانات الخاصة للتدريب زادة الدقة حتى وصلت الى 93% [12].

### 3- الأدوات والتقنيات المستخدمة

هنا سوف يتم شرح طريقة عمل الشبكات العصبية الاصطناعية والانتفاخية العميقة وستتطرق إلى الأدوات التي تم استخدامها لبناء النماذج وتدريبها. مثل الطبقات والخوارزميات وكذلك التقنيات التي تم استخدامها في تطور النظام.

### 1.3 الشبكات العصبية الاصطناعية ANN

هي شبكة عصبية اصطناعية تحاكي آلية عمل الدماغ وتم استلهامها من الدماغ والأعصاب البشرية كما في الشكل التالي [5]:



الشكل (1): يوضح الشبكة العصبية

نلاحظ في الشكل (1)، الآتي :

- إشارات الدخل  $Input(x_1, x_2, \dots, x_n)$  والتي تمثل إشارة المدخلات للخلية العصبية
  - الأوزان  $Weight(w_1, w_2, \dots, w_n)$  تشير الأوزان هنا إلى الترابط ما بين العناصر
  - تابع التنشيط أو التفعيل (Activation Function) وظيفته إخماد العصبون، وجعل نطاقه بين 0 و 1
  - الخرج  $Output(Y_{ij})$  وهو الناتج النهائي للعمليات داخل الشبكة العصبية
- تتكون الشبكة العصبية الاصطناعية (Artificial Neural Networks) (ANNs)، واختصارها (ANNs) من ثلاث طبقات أساسية طبقة الإدخال، الطبقة المخفية وطبقة الإخراج. تتكون كل طبقة من عقد (nodes) أو كما تسمى أيضا بالخلايا العصبية (neurons) وكل عقدة تكون مترابطة مع العقدة في الطبقة التالية. يعتمد عدد العقد في طبقة الإدخال والإخراج على الخصائص والسمات الموجودة في مجموعة البيانات. أما بالنسبة للطبقات المخفية فيمكن زيادتها بحسب المشكلة المراد حلها دون قيود.

### 2.3 الشبكات العصبية الالتفافية (Convolutional Neural Networks)

الشبكات العصبية الالتفافية واختصارها (CNNs)، تعتبر شبكات عصبية اصطناعية عميقة، يتم استخدامها في تصنيف الصور Image recognition وكذلك في التعرف على الأشياء Object recognition. تتكون الـ CNNs من أربع طبقات أساسية وهي:

- طبقة الالتفان Convolute Layer وهي الطبقة الأولى والرئيسية في الـ (CNN)، تستخدم في هذه الطبقة ما يسمى نواة Kernel أو مرشح Filter يتم عن طريقه مسح الصورة ومن ثم جمع البكسل Pixel واستخراج القيم النهائية كميزات Features وخارطة ميزات Features Map والشكل التالي يوضح آلية الـ Convolute Layer [13]:

$$\begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 0 & 0 & -1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 0 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline 1 & 0 & -1 \\ \hline 0 & 0 & -1 \\ \hline 0 & 0 & -1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \sum \begin{array}{|c|c|} \hline 0 & 2 \\ \hline -1 & -1 \\ \hline \end{array}$$

المرشح Filter or  
الصورة المدخلة (Input)  
الخرج أو خارطة السمات  
Output or Features map

الشكل (2) : يوضح آلية عمل طبقة الالتفان

- طبقة التجميع Pooling Layer غالبا ما تأتي هذه الطبقة بعد طبقة الالتفان وتقوم هذه الطبقة بتقليل أبعاد الصورة وتتكون هذه الطبقة من عدة أنواع منها:
  1. طبقة Max pooling تأخذ أعلى قيمة
  2. طبقة Average Pooling تأخذ متوسط القيم
  3. طبقة Min Pooling تأخذ أقل قيمةالشكل التالي يوضح آلية عمل طبقة التجميع:

## العدد السابعون / ديسمبر / 2023

Max p		Min p																								
<table border="1"><tr><td>6</td><td>8</td></tr><tr><td>3</td><td>4</td></tr></table>	6	8	3	4	<table border="1"><tr><td>1</td><td>1</td><td>2</td><td>4</td></tr><tr><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	1	1	2	4	5	6	7	8	3	2	1	0	1	2	3	4	<table border="1"><tr><td>1</td><td>2</td></tr><tr><td>1</td><td>0</td></tr></table>	1	2	1	0
6	8																									
3	4																									
1	1	2	4																							
5	6	7	8																							
3	2	1	0																							
1	2	3	4																							
1	2																									
1	0																									
	Avg p																									
	<table border="1"><tr><td>3.25</td><td>5.25</td></tr><tr><td>2</td><td>2</td></tr></table>	3.25	5.25	2	2																					
3.25	5.25																									
2	2																									

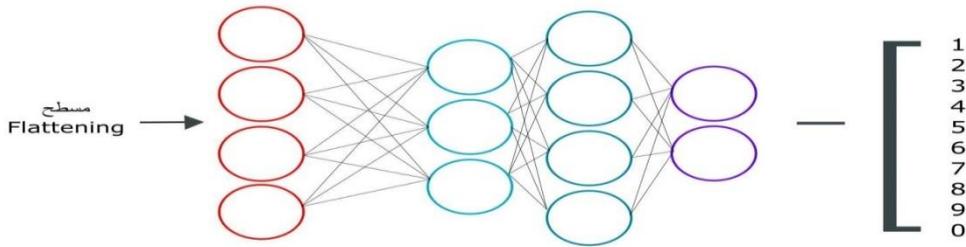
الشكل (3): يوضح آلية عمل طبقة التجميع

- نستخدم في هذا البحث أعلى قيمة الـ Max pooling لأنها أكثر شيوعاً واستخداماً وتفيد في تقليل الأبعاد أكثر.
- طبقة التسطيح Flatten Layer مهمة هذه الطبقة هي تسطيح البيانات من مصفوفة Array إلى متجه Vector وبذلك تصبح البيانات مسطحة كأنبوب.



الشكل (3): يوضح آلية عمل طبقة التسطيح

- طبقة الاتصال بالكامل Fully Connected وهي الطبقة الأخيرة والتي يتم فيها ربط العصبونات ببعضها، من أجل إخراج النتيجة النهائية.

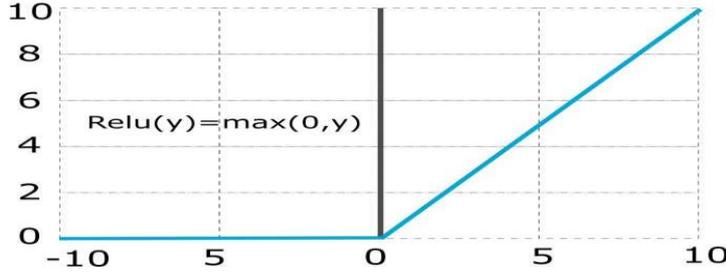


الشكل (4) يوضح آلية عمل طبقة الاتصال بالكامل

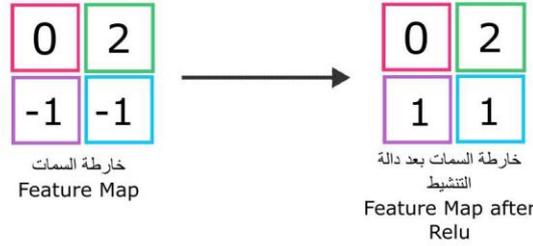
- طبقة التنشيط نوع Rectified Linear Unit هذه الطبقة غالباً ما تستخدم مع طبقة الالتفاف وتسمى اختصاراً Relu تقوم بتحويل المدخلات التي تم ترشيحها أو فلترتها، إلى قيمة معينة ومن ثم إرسالها للطبقة التالية. نطاق هذه الطبقة هو (0,1) حيث تقوم هذه الطبقة بإخراج القيمة كما هي إذا كانت موجبة. أما إذا كانت سالبة فالناتج يكون هنا صفر.

## العدد السابعون / ديسمبر / 2023

تعتبر هذه الطبقة أو كما تسمى ايضا دالة أكثر استخداما في الطبقات العميقة وذلك لأنها تقلل من وقت التدريب. الشكل التالي يوضح دالة حساب دالة التنشيط Relu و الشكل(6) يوضح آلية عمل الدالة Relu.

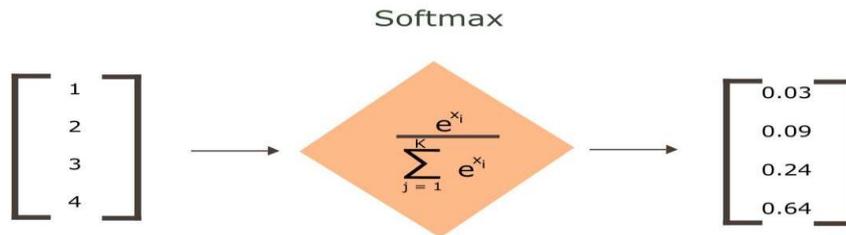


الشكل(5): يوضح حساب دالة التنشيط Relu



الشكل(6): دالة التنشيط Relu

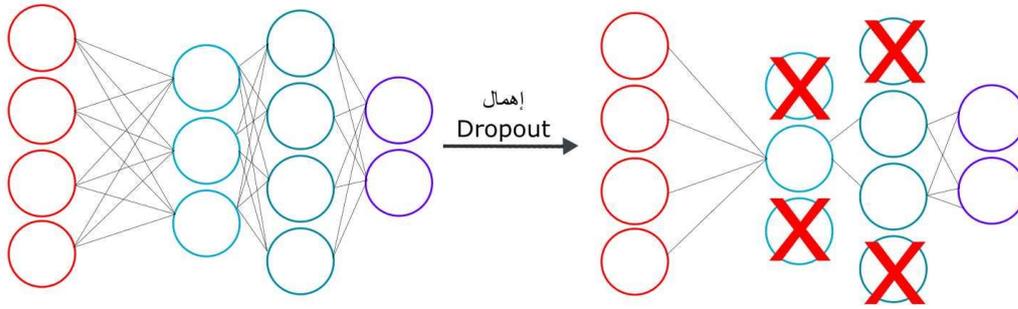
- دالة التنشيط Softmax Function تستخدم في الطبقة الأخيرة وفي التصنيف المتعدد وتقوم هذه الدالة بتحويل متجه مكون من قيم حقيقية . تمثل  $K$  ، الى متجه بقيم حقيقية ومجموعها يساوي 1 أي تقوم بتحويل أي قيمة سواء سالبة أو موجبة أو صفرية الى قيمة محصورة بين (0,1) ويتم اعتبار هذه القيم كاحتمالات Probabilities. الشكل (7)، يوضح آلية عمل الـ(Softmax):



الشكل(7): دالة التنشيط Softmax

## العدد السابعون / ديسمبر / 2023

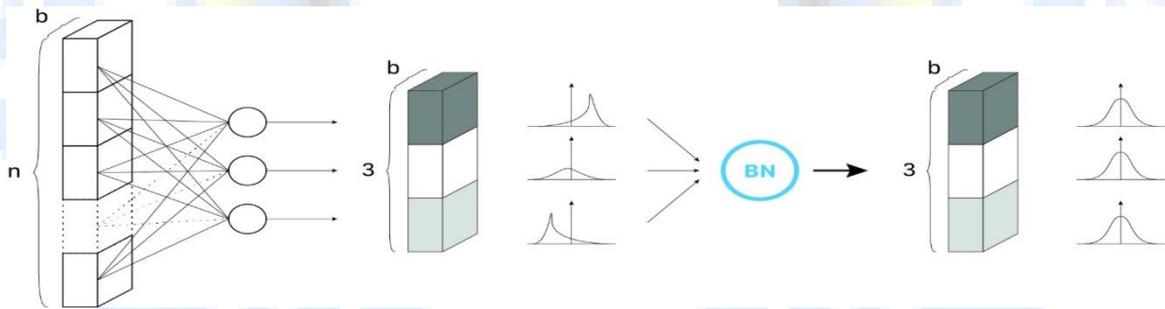
- طبقة الإهمال Dropout Layer طبقة تستخدم أثناء تدريب الشبكة، وذلك لتجنب مشكلة تعتبر شائعة في الشبكات العصبية تسمى Overfitting وآلية عمل طبقة الإهمال تتمثل في إيقاف أو تعطيل بعض العصبونات في الطبقات المخفية Hidden Layers أثناء التدريب Training ويتم اختيار العصبونات بشكل عشوائي كما موضح في الشكل (8).



الشكل (8): طبقة الإهمال

- طبقة التطبيع Batch Normalization هي عبارة عن طبقة يمكن إدراجها ضمن طبقات الشبكة العصبية وظيفتها هذه الطبقة هي تسوية بيانات الدخل وجعلها في نطاق بين (0,1) بحيث يجعلها متمركزة في الجزء الخطي لدالة التنشيط. تجعل هذه الطبقة الشبكة أسرع وقادرة على التعميم بشكل أكبر كما في الشكل (9) يوضح عمل الـ Batch

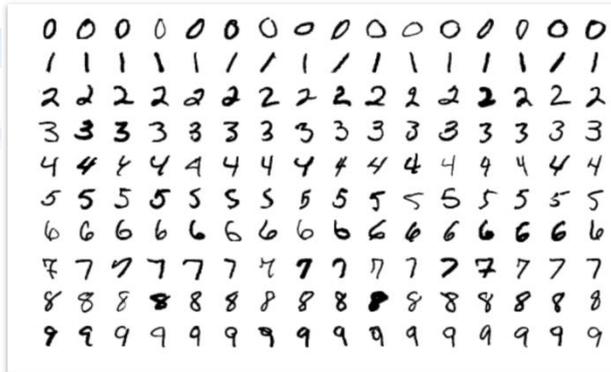
Normalization .



الشكل (9): يوضح طبقة الـ Batch Normalization

### 3.3- قاعدة البيانات المستخدمة هي Modified National Institute of Standards and Technology database (MNIST)

هي مجموعة بيانات Data Set لأرقام مكتوبة بخط اليد وتم قياسها بحجم معين وهو  $28 \times 28$  بكسل. يتم استخدامها لتدريب واختبار النماذج المختلفة Models في معالجة الصور. تتكون MNIST من 70,000 صورة مقسمة إلى 60,000 للتدريب Training و 10,000 للاختبار Testing [7].



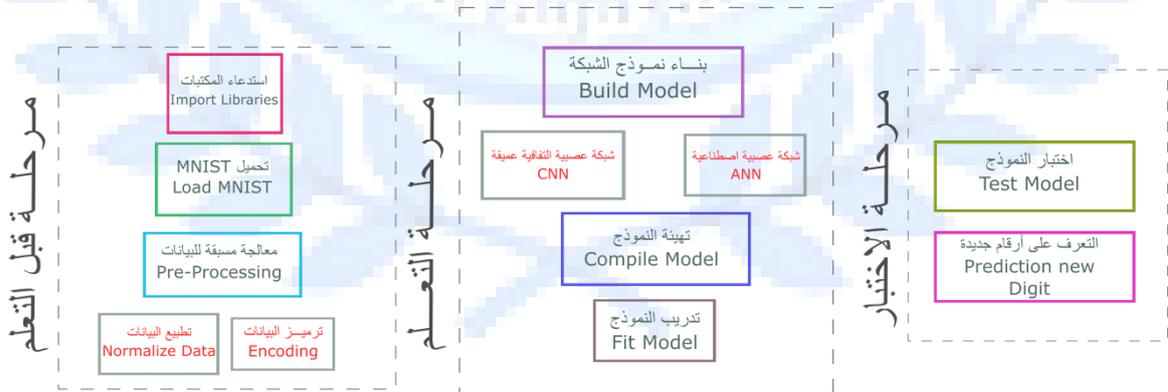
الشكل (10): مجموعة البيانات الخاصة بالأرقام المكتوبة بخط اليد

#### 4 المنهجية

يتناول هذا الجزء شرح مفصل لمنهجية البحث التي توضح جميع المراحل التي تم اتباعها في هذا البحث.

#### 1.4 منهجية البحث

توضح منهجية البحث المراحل التي تم اتباعها في هذا البحث من مرحلة إعداد البيانات وهي من مرحلة قبل التعلم الى مرحلة الاختبار والتعرف. الشكل (11) يوضح هذه المراحل.



الشكل (11): يوضح مراحل البحث

## العدد السابعون / ديسمبر / 2023

### 1.1.4- مرحلة قبل التعلم (مرحلة إعداد البيانات)

وهي المرحلة التي يتم فيها الإعداد الأولي للبيانات بالإضافة الى استدعاء البيانات وتحميل MNIST .

- استدعاء المكتبات:

الشكل (12) يوضح استدعاء المكتبات لـ (ANN):

```
#import libraries
import keras
from keras.models import Sequential,Model
from keras.layers import Dense, Flatten, Dropout, BatchNormalization
from keras.datasets import mnist
from keras.utils.np_utils import to_categorical
from keras.utils.vis_utils import model_to_dot
import matplotlib.pyplot as plt
import numpy as np
```

الشكل(12) استدعاء مكتبات لـ ANN

والشكل(13) يوضح استدعاء مكتبات لـ CNN

```
#import libraries
import keras
from keras.models import Sequential
from keras.layers import Conv2D , MaxPool2D
from keras.layers import Dense, Flatten ,Dropout ,BatchNormalization
from keras.datasets import mnist
from keras.utils.np_utils import to_categorical
from keras.utils.vis_utils import model_to_dot
import matplotlib.pyplot as plt
import numpy as np
```

الشكل(13) استدعاء مكتبات لـ CNN

لا يوجد فرق في المكتبات فقط أستخدمنا نفس المكتبات لكلا الشبكتين والفرق الوحيد هو هيكلية كل شبكة فمثلا شبكة CNN تختلف عن ANN في البنية فهي تملك طبقات مثل Conv layer Max-pool Layer .

- تحميل قاعدة البيانات MNIST

ثم يتم بعد الخطوة السابقة تحميل مجموعة البيانات التي سندرب عليها الشبكات ونختبرها عن طريق تحميلها أو استدعائها مباشرة من الموقع الأصلي ويتم تقسيمها لمجموعتين مجموعة تدريب ( $X_{train}$ ) و ( $Y_{train}$ ) ومجموعة اختبار ( $X_{test}$ ) و ( $Y_{test}$ ) كما في الشكل(14):

```
(x_train,y_train),(x_test,y_test)=mnist.load_data()

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [=====] - 0s 0us/step
11501568/11490434 [=====] - 0s 0us/step
```

الشكل(14):تحميل MNIST

• المعالجة المسبقة Pre-Processing

إعادة تشكيل البيانات (Reshape)، كما الشكل(15):

```
x_train= x_train.reshape(x_train.shape[0],28,28,1)  
x_test= x_test.reshape(x_test.shape[0],28,28,1)
```

الشكل(15): يوضح Reshape

إضافة بعد رابع لبيانات صور التدريب والاختبار حيث يصبح شكل بيانات التدريب (60000,28,28,1) وشكل بيانات الاختبار(10000,28,28,1)، والرقم (1)، توضح أن الصور ذات تدرج رمادي حتى يتعرف عليها النموذج.

• تطبيع البيانات Normalize

نقوم بتحويل نوع البيانات من Integer الى Float ثم نقوم بقسمة كل قيمة بكسمل على 255.

```
x_train =x_train.astype('float32')  
x_test=x_test.astype('float32')  
x_train/= 255.0  
x_test/= 255.0
```

الشكل (16) تطبيع البيانات

ترميز البيانات One Hot encoder

تحويل صيغة البيانات للمجموعات سواء مجموعة التدريب أو الاختبار من أعداد صحيحة (int) الى مصفوفة فئة ثنائية binary

class matrix الشكل (17) يوضح one hot encoder

0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

الشكل (17) ترميز البيانات

## العدد السابعون / ديسمبر / 2023

الشكل (18) يوضح الكود الخاص بالترميز، وقد استخدمنا دالة categorical

```
n_class = 10
y_train = to_categorical(y_train,n_class)
y_test = to_categorical(y_test,n_class)
print(y_train[0])

[0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
```

الشكل (18) ترميز البيانات

5- مرحلة التطبيق والاختبار

1.5 مرحلة التطبيق

1.1.5 اولا مرحلة التطبيق قمنا ببناء نموذج Model للشبكة ANN كما موضح في الشكل (19)

```
model = Sequential()
model.add(Flatten(input_shape=(28,28)))

model.add(Dense(256,activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(200, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(128,activation = 'relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(10,activation='softmax'))
```

الشكل (19) بنية ANN

كما هو موضح بالشكل يتكون النموذج أساسا من خمس طبقات أساسية بدءا من طبقة (Flatten) بالإضافة الى الطبقات الأخرى.

الطبقة الأولى تمثل طبقة الإدخال والتي تعتبر نواة النموذج.

الطبقة الثانية تعبر عن طبقة الاتصال بالكامل (Dense) وقد ضمنا لها 256 عقدة neurons مع دالة تنشيط من نوع

Relu وأتبعناها بطبقتين تحسين وهما Batch Normalization و Dropout بنسبة إهمال 50%.

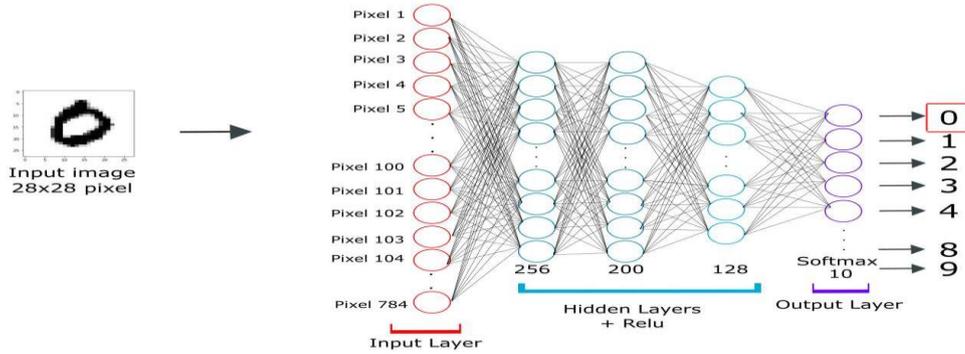
والطبقة الثالثة والرابعة نفس الطبقة السابقة ولكن واحدة ب200 عقدة و الأخرى ب128 عقدة فقط

أما الطبقة الأخيرة فهي طبقة الإخراج والتي تتضمن عدد الفئات Classes وعددها 10 فئات بعدد الأرقام من 0 الى 9 مع

دالة تنشيط من نوع Softmax.

العدد السابعون / ديسمبر / 2023

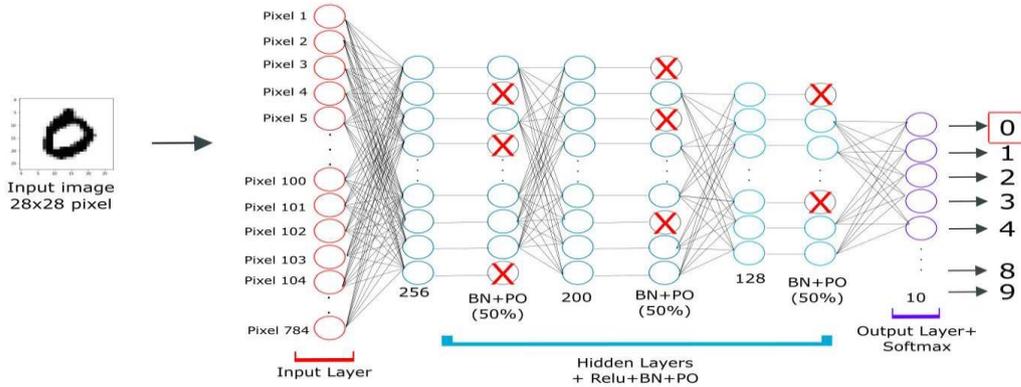
الشكل (20) يوضح هيكلية الشبكة بدون Batch Normalization و Dropout



الشكل (20) هيكلية ANN بدون (BN)، (PO)

BN و PO هي تقنية مستخدمة على نطاق واسع في مجال التعلم العميق. وظيفتها تحسن سرعة التعلم للشبكات العصبية ويوفر التنظيم وتجنب الإفراط في التجهيز.

والشكل (21) يوضح هيكلية الشبكة مع Batch Normalization و Dropout



الشكل (21) هيكلية ANN مع (BN)، (PO)

2.1.5 بناء نموذج CNN، الشكل (22) يوضح بنية الشبكة

كما هو موضح الكود، يتكون نموذج CNN من ست طبقات أساسية، بالإضافة الى طبقات BN و PO.

```

model.add(Conv2D(64, kernel_size=2, padding = 'same', strides=1, activation = 'relu', input_shape = (28, 28, 1)))
model.add(MaxPooling2D(pool_size=(2, 2), strides=2))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Conv2D(32, kernel_size=2, strides=1, padding = 'same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=2))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Conv2D(128, kernel_size=2, padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2), strides=2))
model.add(BatchNormalization())
model.add(Dropout(0.5))

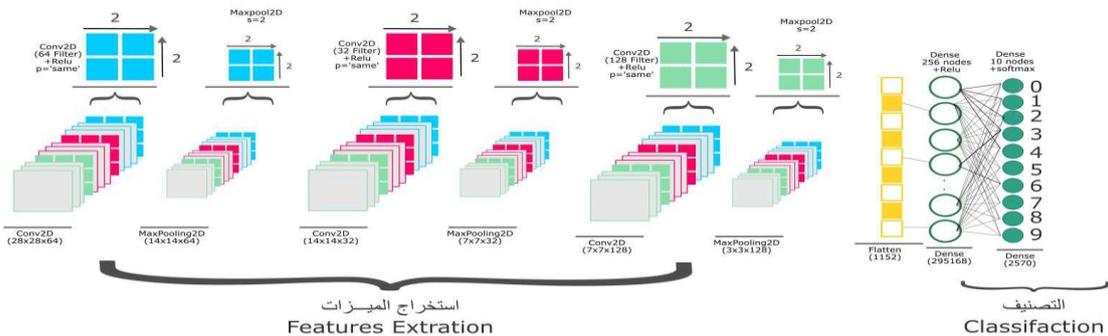
model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

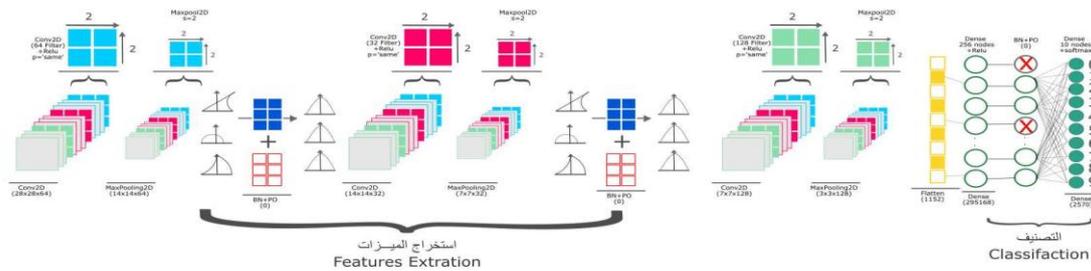
model.add(Dense(n_class, activation='softmax'))
    
```

### الشكل (22) بنية CNN

تمثل الطبقة الأولى طبقة الإدخال مكونة من طبقة التفاف بـ 64 مرشح Filter بأبعاد 3x3. بالإضافة الى الحشو نوع Same بخطوة واحدة، ودالة تنشيط من نوع Relu ويتبعها طبقة تجميع ذات أبعاد 2x2. والعنصر الأساسي وهو Input\_Shape=(28,28,1) والذي يمثل المدخل الذي يعتبر نواة النموذج. في الطبقة الثانية تم استخدام 32 مرشح Filter بأبعاد 3x3. والطبقة الثالثة تم استخدام 128 مرشح بأبعاد 3x3. في الطبقة الرابعة طبقة تسطيح لتحويل البيانات من مصفوفة الى متجه وفي الطبقة الخامسة استخدمنا طبقة اتصال بالكامل بـ 256 عقدة مع دالة تنشيط Relu. أما الطبقة الأخيرة فهي طبقة الإخراج وفيها عدد الفئات n\_class بعدد الأرقام المراد التعرف عليها وهي 10 مع دالة تنشيط Softmax. الشكل (23) يوضح هيكلية CNN بدون BN و PO.



الشكل (23) هيكلية CNN بدون BN و PO



الشكل(24) هيكلية CNN مع BN و PO

### 3.1.5 تهيئة النموذج Model Compile

هذه المرحلة تأتي قبل عملية التدريب والغرض منها هو تهيئة أو ترجمة النموذج حتى يتمكن من تدريبها. الشفرة المكتوبة بالاسفل توضح كود تهيئة النموذج.

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

كما هو موضح في الشفرة فقد استخدمنا دالة categorical\_crossentropy لأنها تعتبر الأنسب في حالة التصنيف المتعدد multiclass كما استخدمنا خوارزمية التحسين adam وأخيرا عملية تتبع للدقة.

### 4.1.5 تدريب النموذج (Fit Model).

تحديد البيانات التي نريد أن ندرّب الشبكة عليها وهي (x\_train,y\_train)، ومن ثم تحديد حجم الدفعة وهي = 35 واختيار فترات التدريب وهنا اخترنا فترات مختلفة ونحدد بيانات التحقق من الصحة بنسبة 30 بالمئة . الشفرة المكتوبة بالاسفل توضح كود عملية التدريب النموذج.

```
hist =model.fit(x_train,y_train, validation_split=0.3,epochs=100, batch_size=35,verbose=1)
```

### 2.5 اختبار النموذج Test Model

في هذه المرحلة نستدعي النموذج الذي تم حفظه ونقوم بالتعرف على أرقام مجموعة التدريب من خلال رقم الفهرس.

### 1.2.5 التعرف على أرقام جديدة

الشكل (25) يوضح لنا المعالجة المسبقة للصور المدخلة وذلك بأول خطوة نقوم بها وهي استدعاء مكتبات معالجة الصور Preprocessing وتمثل الخطوات التي بعدها جعل الصور رمادية وتحديد حجمها (28,28) ثم تحويلها من (int) الى (float) واخيرا قسمتها على 255 حتى تكون في نطاق محدد.

## العدد السابعون / ديسمبر / 2023

```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.models import load_model

def load_image(filename):
    img = load_img(filename, grayscale=True, target_size=(28,28))
    img = img_to_array(img)
    img = img.reshape(1,28,28)
    img = img.astype('float32')
    img = img / 255
    return img
```

الشكل (25) يوضح معالجة الصور

### 6 النتائج و الاستنتاجات

في هذا الجزء نستعرض نتائج كل من ANN و CNN ومقارنتها بناء على الدقة والأداء والوقت بالإضافة الى تغيير عدد الفترات كذلك مقارنة مدى فعاليتها باستخدام طبقات BN+PO وبدون استخدام طبقات BN+PO.

### 1.2.6 النتائج بدون (BN+PO)

الان الجداول التالية يوضح الفروقات في حال عدم إضافة BN+PO

- عدد الفترات (Epochs= 5,10,25)
- حجم الحزمة (Batch size = 64)

الجدول (1) نتائج ANN

عدد الفترات (Epochs)			الأرقام
25	10	5	
%24 – 6	%37 – 4	%66 – 5	0
%61 – 8	%54 – 1	%49 – 5	1
%92 – 2	%42 – 2	%21 – 8	2
%42 – 1	%58 – 1	%37 – 1	3
%79 – 4	%88 – 4	%72 – 4	4
%99 – 5	%61 – 4	%60 – 5	5
%50 – 5	%57 – 4	%56 – 5	6

العدد السابعون / ديسمبر / 2023

%61 – 7	%93 – 1	%44 – 1	7
%54 – 4	%68 – 4	%45 – 4	8
%42 – 7	%72 – 1	%44 – 4	9
%94	%92	%89	الدقة
1 د 44 ث	1 د 43 ث	33 ث	الوقت المنجز

الجدول (2) نتائج CNN

عدد الفترات (Epochs)			الأرقام
25	10	5	
%98 – 1	%38 – 1	%39 – 1	0
%99 – 1	%99 – 1	%99 – 1	1
%49 – 1	%60 – 3	%49 – 1	2
%42 – 1	%85 – 3	%96 – 1	3
%79 – 4	%71 – 1	%63 – 4	4
%99 – 5	%93 – 5	%54 – 5	5
%50 – 5	%34 – 5	%53 – 9	6
%61 – 7	%96 – 3	%65 – 7	7
%54 – 4	%58 – 5	%53 – 1	8
%42 – 7	%28 – 1	%82 – 9	9
%96	%94	%91	الدقة
24 د	12 د 22 ث	7 د 23 ث	الوقت المنجز

العدد السابعون / ديسمبر / 2023

نلاحظ ان في هذا الاختبار نتائج CNN افضل من ANN.  
• الان تم تغير حجم الحزمة الى 35 .

الجدول(3)نتائج ANN

عدد الفترات (Epochs)			الأرقام
25	10	5	
%71 - 4	%46 - 4	%34 - 4	0
%59 - 8	%40 - 6	%58 - 1	1
%73 - 2	%77 - 2	%82 - 2	2
%63 - 1	%67 - 1	%85 - 1	3
%90 - 4	%78 - 4	%88 - 4	4
%66 - 4	%63 - 6	%60 - 5	5
%75 - 4	%23 - 5	%33 - 5	6
%48 - 7	%63 - 7	%66 - 1	7
%44 - 4	%61 - 1	%33 - 1	8
%29 - 1	%31 - 7	%34 - 1	9
%94	%91	%89	الدقة
4 د 23 ث	1 د 30 ث	48 ث	الوقت المنجز

العدد السابعون / ديسمبر / 2023

الجدول (4) نتائج CNN

عدد الفترات (Epochs)			الأرقام
25	10	5	
%46 - 3	%74 - 0	%95 - 1	0
%99 - 1	%99 - 1	%99 - 1	1
%37 - 5	%69 - 2	%37 - 3	2
%61 - 3	%88 - 3	%89 - 1	3
%46 - 4	%77 - 4	%96 - 1	4
%77 - 5	%30 - 5	%85 - 5	5
%52 - 1	%74 - 9	%48 - 1	6
%59 - 3	%63 - 7	%64 - 3	7
%45 - 1	%64 - 9	%57 - 9	8
%33 - 9	%56 - 9	%88 - 1	9
%96	%94	%90	الدقة
24 د 31 ث	22 د 13 ث	23 د 7 ث	الوقت المنجز

مع تغيير حجم الحزمة الى 35 نقصت دقت CNN قليلا ولكن لازلت افضل من ANN

1.2.6 النتائج مع (BN+PO)

الجدول التالية توضح الفروقات مع استخدام BN+PO ومع الشروط التالية:

- عدد الفترات (Epochs = 5, 10, 25)
- حجم الحزمة (Batch size = 64)

العدد السابعون / ديسمبر / 2023

الجدول (5) نتائج ANN

عدد الفترات (Epochs)			الأرقام
25	10	5	
%45 - 5	%35 - 5	%45 - 8	0
%72 - 8	%91 - 8	%81 - 8	1
%58 - 2	%36 - 5	%69 - 2	2
%42 - 2	%48 - 1	%56 - 1	3
%39 - 7	%36 - 2	%54 - 7	4
%85 - 5	%88 - 5	%59 - 5	5
%73 - 8	%81 - 5	%35 - 5	6
%53 - 1	%53 - 3	%53 - 7	7
%45 - 4	%70 - 9	%39 - 1	8
%61 - 3	%37 - 3	%54 - 8	9
98	97	96	الدقة
د 2	د و 22 ث	ث 24	الوقت المنجز

نلاحظ ان في جدول (5) عند استخدام BN+PO زادت دقة ANN وكذلك مع تقنية CNN كما موضح في جدول (6).

العدد السابعون / ديسمبر / 2023

الجدول (6) نتائج CNN

عدد الفترات (Epochs)			الأرقام
25	10	5	
%35 - 0	%45 - 3	%51 - 3	0
%56 - 1	%98 - 1	%77 - 1	1
%92 - 5	%67 - 1	%43 - 3	2
%89 - 3	%64 - 1	%61 - 3	3
%47 - 7	%91 - 4	%72 - 4	4
%94 - 5	%43 - 3	%74 - 3	5
%68 - 5	%40 - 2	%27 - 7	6
%53 - 3	%57 - 3	%56 - 7	7
%73 - 5	%85 - 9	%63 - 9	8
%94 - 9	%95 - 9	%69 - 7	9
%99	%98	%97	الدقة
د 22	د 22 ث	د 5	الوقت المنجز

• في هذه التجربة قم بتغيير حجم الحزمة (Batch size = 35)

العدد السابعون / ديسمبر / 2023

الجدول (7) نتائج ANN

عدد الفترات (Epochs)			الأرقام
25	10	5	
%11 - 0	%45 - 6	%50 - 5	0
%11 - 0	%99 - 8	%35 - 6	1
%11 - 0	%91 - 2	%36 - 2	2
%12 - 7	%41 - 1	%72 - 1	3
%11 - 0	%98 - 3	%25 - 3	4
%11 - 0	%83 - 5	%82 - 5	5
%12 - 0	%39 - 6	%36 - 5	6
%12 - 0	%50 - 7	%29 - 1	7
%12 - 0	%71 - 4	%23 - 1	8
%11 - 0	%56 - 3	%22 - 7	9
%98	%98	%96	الدقة
د 2	د 1 و 23 ث	د 3 و 21 ث	الوقت المنجز

العدد السابعون / ديسمبر / 2023

الجدول (8) نتائج CNN

عدد الفترات (Epochs)			الأرقام
25	10	5	
%70 - 7	%43 - 7	%43 - 3	0
%69 - 1	%62 - 1	%84 - 1	1
%79 - 3	%48 - 7	%55 - 3	2
%83 - 3	%46 - 2	%68 - 1	3
%72 - 4	%98 - 4	%66 - 4	4
%89 - 9	%59 - 5	%50 - 3	5
%55 - 1	%33 - 3	%26 - 5	6
%55 - 3	%89 - 7	%82 - 3	7
%64 - 9	%63 - 9	%52 - 5	8
%73 - 7	%78 - 9	%34 - 1	9
%99	%98	%97	الدقة
21 د 25 ث	10 د و 19 ث	5 د 22 ث	الوقت المنجز

نلاحظ عند تغير حجم الحزمة الى 35 Batch Size لا يوجد الفرق كبير بين النتائج لكل التقنيتين. وبالنسبة للنتائج تم ملاحظة ان التقنية CNN افضل نتائج من ANN وخصوصا عندما تم استخدام BN+PO .

7 الاستنتاجات

- ✓ الدقة: CNN هي الافضل في الدقة حيث وصلت الى %99 مع عد الفترة 35.
- ✓ الأداء : من الجداول الموضحة أعلاه تبين أن CNN استطاعت أن تتعرف على اغلب الارقام من أصل 10 أرقام بينما ANN استطاعت أن تتعرف على حوالي نصف الارقام.

## العدد السابعون / ديسمبر / 2023

- ✓ الوقت : بالنسبة لوقت التدريب فتعتبر ANN أسرع من CNN.
- ✓ عدم التعلم : تعتبر ANN أكثر عرضة لفرط التجهيز مقارنة بي CNN أقل عرضة منها.
- ✓ تبين كذلك أن طبقات التحسين BN+PO استطاعت أن تجنبنا فرط التجهيز بنسبة جيدة كما استطاعت من تحسين الشبكات للتعلم.

### 8- الخاتمة

تم بعون الله انشاء نظام اختبار ما بين تقنية من تقنيات تعليم الالة في مجال الذكاء الاصطناعي. وتم الحصول على نتائج جيدة وضحت لدينا أداء كل تقنية وما يميز كل تقنية عن الأخرى. مما تبين لنا ان تقنية CNN استطاعت تميز الارقام المكتوبة بخط اليد افضل من تقنية ANN خاصة عند اضافة طبقات BN+PO التي ساهمت في تحسين أداء التقنيتين. ولكن من ناحية الزمن فان تقنية ANN هي الافضل.

### 9- التوصيات المستقبلية

- لتطوير هذه الدراسة وتحسين أداؤها في المستقبل ننصح بالاتي:
- استخدام قاعدة بيانات اكبر وهي التي تعطي للبرنامج مجال تعلم أكثر منها تكون النتائج افضل.
- استخدام طبقات تحسين بما يتناسب مع حجم الشبكات (عدد الطبقات)
- توفير وقت كافي للشبكات حتى تتعلم وذلك بزيادة عدد الفترات

المراجع

1. Jagtap, V. N., & Mishra, S. K. (2014). Fast efficient artificial neural network for handwritten digit recognition. *International Journal of Computer Science and Information Technologies*, 5(2), 2302-2306.
2. Saeed, A. M. (2015). Intelligent handwritten digit recognition using artificial neural network. *Int. Journal of Engineering Research and Applications*, 5(5), 46-51.
3. Sakshica, D., & Gupta, K. (2015). Handwritten digit recognition using various neural network approaches. *Int. J. Adv. Res. Comput. Commun. Eng*, 4(2).
4. Roshni Musaddi.2018 . Handwritten Digit Recognition using Neural Network. *Journal of Network Communications and Emerging Technologies (JNCET)*  
Volume 8, Issue 10. ISSN: 2395-5317
5. Hamid, N. A., & Sjarif, N. N. A. (2017). Handwritten recognition using SVM, KNN and neural network. arXiv preprint arXiv:1702.00723.
6. Abdulrazzaq, M. B., & Saeed, J. N. (2019, April). A comparison of three classification algorithms for handwritten digit recognition. In *2019 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 58-63). IEEE.
7. Tabik, S., Peralta, D., Herrera-Poyatos, A., & Herrera Triguero, F. (2017). A snapshot of image pre-processing for convolutional neural networks: case study of MNIST.
8. Ghosh, M. M. A., & Maghari, A. Y. (2017, October). A comparative study on handwriting digit recognition using neural networks. In *2017 international conference on promising electronic technologies (ICPET)* (pp. 77-81). IEEE.
9. Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved handwritten digit recognition using convolutional neural networks (CNN). *Sensors*, 20(12), 3344.
10. Vinjit, B. M., Bhojak, M. K., Kumar, S., & Nikam, G. Implementation of Handwritten Digit Recognizer using CNN.
11. Pashine, S., Dixit, R., & Kushwah, R. (2021). Handwritten Digit Recognition using Machine and Deep Learning Algorithms. *arXiv preprint arXiv:2106.12614*.
12. Ghazal, T. M. (2022). Convolutional neural network based intelligent handwritten document recognition. *Computers, Materials & Continua*, 70(3), 4563-4581.

العدد السابعون / ديسمبر / 2023

13. Walter, B. (2023). Analysis of convolutional neural network image classifiers in a hierarchical max-pooling model with additional local pooling. Journal of Statistical Planning and Inference, 224, 109-126.

