Aktuelle Entwicklung und Probleme bei künstlichen neuronalen Netzen in Bezug auf Deep Learning

Mario Neises
Hochschule für Technik und Wirtschaft Berlin

***Zusammenfassung*—Es wird erläutert, für welche Problemstellungen künstliche neuronale Netze einen Nutzen bieten und wie diese grundlegend aufgebaut sind. Die Fortschritte hin zum Deep Learning werden gezeigt und unterschiedliche Probleme untersucht.**

I. Einführung

Heutzutage ist es möglich, dass ein Smartphone gesprochene Fragen beantworten kann oder dass in sozialen Netzwerken Freunde auf Bildern automatisch erkannt und zugeordnet werden. Die Zunahme solcher Entwicklungen beruhen auf einem langen Werdegang in der Forschung und dem Interesse von Unternehmen, das Potenzial davon zu nutzen. Folgende Personen und deren Publikationen sind dazu ein entscheidender Beitrag: Geoffrey E. Hinton[1], Yann LeCun[2] und Yoshua Bengio[3]. Aktuell ist G. Hinton bei Google forschend tätig, Y. LeCun ist bei Facebook Leiter der Forschungsabteilung für künstliche Intelligenz.

A. Anforderungen

Die eingangs erwähnten Probleme sind für Menschen einfach und intuitiv lösbar, für Computer hingegen selten eindeutig zu berechnen. Damit die Fortschritte verstanden werden können, beschreiben folgende Abschnitte die Anforderungen, welche durch die Probleme auftreten. Es wird daraus deutlich, worin die grundlegenden Unterschiede[4] zwischen der Arbeitsweise und Leistungsfähigkeit des menschlichen Gehirns und maschineller Verarbeitung bestehen.

1) Generalisierung: Bei Text-, Sprach- und Bildererkennung wird die Eingabe auf wenige Eigenschaften reduziert. Die Struktur soll aus der ursprünglichen Darstellung gewonnen werden. Durch die Merkmalsreduktion können neue Eingaben nach bereits bekanntem Schema verarbeitet werden. *Faktor- und Hauptkomponentenanalyse (engl.: PCA)* stellen bereits Möglichkeiten da, Eingabedatensätze um Dimensionen zu reduzieren.

2) Lernfähigkeit: Heutzutage werden große Datenmengen erfasst, welche aufgrund der Schnelllebigkeit oder Komplexität nicht mehr mit derart klassischen Methoden analysiert und verarbeitet werden können. Daher ist es notwendig, dass die Erkennung von Merkmalen nicht explizit durch Menschen vorgenommen wird. Analog zum menschlichen Lernen sollen die relevanten Eigenschaften und Zusammenhänge durch Betrachten von verschiedenen Beispielen oder Datensätzen verstanden werden.

3) Fehlertoleranz: Eine verrauschte, gestörte oder veränderte Eingabe soll ein Zuordnen nicht beeinträchtigen, sofern die wesentlichen Merkmale davon nicht beeinflusst sind. Erst wenn die Daten zu stark gestört sind und die kausale Beziehung verloren geht, wird eine Erkennung ungenau.

4) Parallelisierbarkeit: Biologische neuronale Strukturen sind in der Lage viele Arbeitsschritte parallel zu erledigen. Dadurch ergibt sich die Leistungsfähigkeit, da zum Beispiel, aus technischer Sicht, Suchprobleme auf vielen kleinen Mengen parallel arbeiten können, statt diese sequenziell zu lösen. Der Vorteil ergibt sich heutzutage, da mit dem Internet ein Parallelrechner mit nahezu unendlicher Kapazität zur Verfügung steht.

Diese Anforderungen werden dem Oberbegriff *Maschinelles Lernen* zugeordnet. Im Rahmen dieser Arbeit wird der Bereich *Deep Learning* behandelt. Andere Ansätze zur Klassifikation oder Regression werden hier nicht näher betrachtet, auch wenn diese Teilbereiche oder ähnliche Probleme abdecken.

II. Komponenten und Methoden Neuronaler Netze

Deep Learning hat sich aus unterschiedlichen Forschungsergebnissen heraus gebildet. Um das besser zu verstehen, werden wesentliche Komponenten, Methoden und Eigenschaften erläutert.

A. Perceptron

Ein künstliches Neuron verarbeitet mehrere Eingaben in einer gewichteten Summe und erzeugt über eine nicht-lineare Aktivierungsfunktion eine Ausgabe.

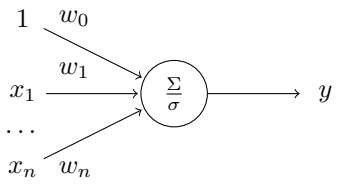

Abbildung 1. Perceptron, bzw. künstliches Neuron. Es verarbeitet die Eingabe $x_1, \dots x_n$, welches z.B. Pixel von einem Bild sein können, mit den Gewichten $w_0, \dots w_n$. Die konstante Eingabe 1 verschiebt die Aktivierungsfunktion entlang der x-Achse.

Ein einfaches Neuron bietet eine Schicht trainierbarer Gewichte und wird als *Single-Layer-Perceptron (SLP)* bezeichnet. Durch das Anpassen der Gewichte können unterschiedliche

[1] https://www.cs.toronto.edu/~hinton/papers.html
[2] http://yann.lecun.com/exdb/publis/index.html
[3] http://www.iro.umontreal.ca/~lisa/publications2/index.php/authors/show/1
[4] Für einen detaillierten Überblick wird auf [1, Kapitel 1] verwiesen, dass die Unterschiede zwischen paralleler und sequentieller Berechnung anhand von Gehirn und Computer diskutiert.

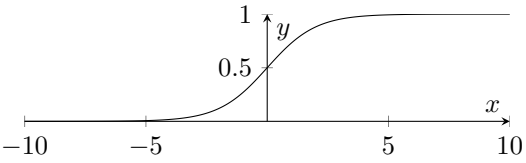

Abbildung 2. Aktivierungsfunktion σ, ordnet der Eingabe x eine Aktivierung zu über die Wahrscheinlichkeitsverteilung (hier: $0 \leqslant y \leqslant 1$).

Klassifikationen vorgenommen werden, da die Abbildung verändert wird. SLP können nur linear-separierbare Probleme lösen. Diese lassen sich im Hyperraum durch eine Hyperebene unterteilen.

Um beliebige Mengen durch Hyperebenen klassifizieren zu können oder um verschiedene Funktionen zu approximieren, werden SLP zu *Multi-Layer-Perceptronen (MLP)* zusammengeschaltet. Dadurch werden mehrere Funktionen überlagert und erlauben das Bestimmen von nicht-linear-separierbaren Problemen. [5]

B. Deltaregel und Backpropagation

SLP und MLP werden über Gewichte trainiert. Diese können einzelne Eingaben oder Verbindungen zu Neuronen verstärken oder schwächen. In der Delta-Regel wird die Differenz zwischen erwarteter und erfolgter Ausgabe gebildet um darüber die Gewichte anzupassen. Beim Backpropagation-Algorithmus wird das *Gradientenabstiegsverfahren* auf mehrschichtige Netze verallgemeinert[5].

Das Verfahren wurde lange Zeit als vielversprechend gesehen, erwies sich jedoch aufgrund diverser Probleme nicht als alleinige Lösung zur Anpassung. Unter anderem dauert die Berechnung bei wenigen Schichten bereits recht lange und kann daher die Fehleranpassungen nicht gut durchpropagieren. Der Ansatz wird jedoch beibehalten und angepasst oder das Verfahren zur Feinjustierung verwendet.

C. Lernparadigmen

Es wird von *überwachtem Lernen* gesprochen, wenn eine erwartete Ausgabe gegeben ist und daher bekannt ist, wie eine Eingabe zu klassifizieren ist. Diese Werte können durchaus auf einem eingegrenzten Bereich bestimmt werden um ein Netz darauf zu trainieren. Dennoch lässt sich das nicht unbedingt mit den Anforderungen vereinbaren, da Netze selbstständig Merkmale extrahieren und auch Eingaben ohne Trainingsbeispiele erkennen sollen. Neben diesem *unüberwachten Lernen* wird insbesondere im Bereich Deep Learning versucht, *semi-überwachtes Lernen* zu nutzen. Das bedeutet, dass das Training selbstständig stattfindet und die Ausgabe einer oder mehrere vordefinierte *Kennzeichnungen (engl.: labels)* mit einer bedingten Wahrscheinlichkeit zugeordnet werden kann.

[5] Unter http://home.htw-berlin.de/~s0549377/alias/NN/ sind verschiedene grafische Modelle verfügbar, die ein intuitives Verständnis von AND- und XOR-Problem sowie Gradientenabstieg vermitteln.

D. Hopffieldnetze und Energiefunktion

Es gibt diverse Netztopologien, welche sich in der Anordnung und Verbindung der Neuronen unterscheidet. Im Gegensatz zu Feed-Forward-Netzen mit gerichteten Verbindungen zu nachfolgenden Neuronen (vergleiche Abbildung 1), sind *Hopffieldnetze* vollständig miteinander verbunden. Eine *Energiefunktion*[6] zeigt an, wie stabil eine potentielle Lösung ist. Aufbauend auf dem Gradientenabstieg können bei großen Fehlern lokale Minima verlassen werden. Da Hopffieldnetze durch die Rückkopplung sich selbst beeinflussen, repräsentieren diese die Ausgabe in ihrem Netzzustand. Daher kann über die Fitnessfunktion bewertet werden, wann das Netz und daher die Ausgabe stabil ist.

III. Deep Learning: Bedeutung und Abgrenzung

Verallgemeinert wird unter Deep Learning die *verkettete Anwendung von Methoden zur Merkmalsextraktion* verstanden. Ziel ist es, *Merkmale oder Eigenschaften (engl. Features)* selbstständig und ohne vorgegebene Logik über die Struktur der Daten zu gewinnen.

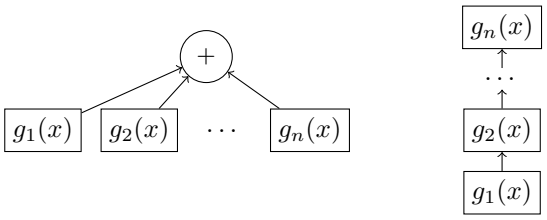

Abbildung 3. Die Funktionen $g_1 \ldots g_n$ entsprechen einzelnen SLP oder MLP. Links: flache Netze verbinden Funktionen, hier über die Summe: $f \approx \sum_i{}^n g_i$ Rechts: tiefe Netze verketten Funktionen: $f \approx g_1(g_2(\ldots g_n))$

Anhand Abbildung 3 ist erkennbar, dass flache Netze in die Breite und tiefe Netze über mehrere Ebenen wachsen. Damit tiefe Netze effektiv trainiert werden können, werden Lernverfahren wie in II-B nur über einzelne MLP ausgeführt. In jeder Ebene können Merkmale extrahiert werden, folglich in der nächsten Ebene Merkmalen von Merkmalen.

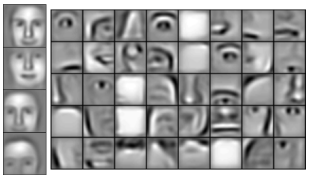

Abbildung 4. Von Gesichtern (links) können pro Ebene Merkmale (rechts) extrahiert werden. Oder aus den Merkmalen können Gesichter generiert werden. Nach [2, Abbildung 3]

A. Modell-Paradigmen

Neben dem klassischen Ansatz der neuronalen Netze können auch mittels der Modellierung als Wahrscheinlichkeitsvertei-

[6] Diese kann als Fitnessfunktion verstanden werden. In der Literatur wird diese als Energiefunktion bezeichnet, basierend auf der physikalischen Idee der *(simulierten) Abkühlung*.

lung gute Ergebnisse erreicht werden[7]. In 2 wurde auf die Möglichkeit bereits hingewiesen und daraus ist erkennbar, dass die Ansätze sich überlagern können und auch Gemeinsamkeiten aufweisen.

1) Neuronale Netze: Diese Netze approximieren Funktionen, welche zur Klassifikation benutzt werden können. Prinzipiell wird die Eingabe auf die Ausgabe eindeutig abgebildet. Die Trennung durch Hyperebene wird in II-A angesprochen.

2) Probabilistische Netze: Die Eingabe wird als Wahrscheinlichkeitsverteilung modelliert. Daher ergibt sich ein Bereich, in dem von einer zuverlässigen Klassifikation ausgegangen wird. In Abbildung 4 ist zu sehen, dass die Daten (Pixel) Korrelationen aufzeigen, welche die Wechselbeziehung der Eingabe untereinander aufzeigt.

B. Netztopologie

In II-D ist erkennbar, dass es unterschiedliche mögliche Strukturen und Verbindungen der Neuronen zueinander gibt. Aufbauend auf [4] lassen sich drei unterschiedliche Typen identifizieren. Einige repräsentative Beispiele gehen auf die Unterschiede und Gemeinsamkeiten sowie auf den aktuellen Stand der Technik ein.

1) Feed-Forward oder generative, unabhängige Klassifikationsmodelle: Diese Netze verbinden die Neuronen vorwärtsgerichtet. Anders betrachtet kann diese mit der unabhängigen Verteilung von Eingaben x beschrieben werden. Sofern diese bestimmt wurde, können daraus neue Eingaben erzeugt werden beruhend auf der Wahrscheinlichkeitsverteilung $p(x, y)$.

2) Feed-Back-Netze oder diskriminative, abhängige Klassifikationsmodelle: Eine Rückkopplung erlaubt es, dass die Ausgabe die Klassifikation der Eingabe wiederum beeinflussen kann. Das bedeutet, dass das bedingte Eintreten von y unter Abhängigkeit von x betrachtet werden kann, nach dem Satz von Bayes als $p(y|x)$ geschrieben.

3) Bi-direktional, hybride Klassifikation: Diese Netze nutzen beide Möglichkeiten. Einerseits wird auf generierten Daten semi-überwacht trainiert, diese Verteilung wird darauf in Abhängigkeit modelliert.

4) Konkrete Entwicklungen: In Abbildung 5 sind zwei verschiedene Umsetzungen von tiefen Netzen zu sehen. *Autoencoder* und *Deep Belief Nets* haben gemeinsam, dass diese, pro Ebene, die Daten in ein neues Netz mit interner Repräsentation überführen. Daher wird von Fantasien der Netze gesprochen. Diese können visualisiert werden und ergeben entsprechende Merkmale, wie in Abbildung 4. Letztendlich ergibt die Ausgabe einen Bereich der zu der Kennzeichnung gehörigen Klasse.[8]

IV. Probleme

Die Arbeiten im Bereich Deep Learning zeigen die Fortschritte bei der Entwicklung. Es werden dazu Lösungen für verschiedene spezifische Probleme vorgestellt. Neben diesen gibt es nun weitere Arbeiten, die sich umfassend mit generellen Problemen befassen.

[7]In [3] wird auch auf die Modell-Betrachtung eingegangen, welche auf Eigenschaften der Mannigfaltigkeit in der Differentialgeometrie beruht. Darauf wird hier nicht weiter eingegangen.

[8]Eine interaktive Visualisierung eines generativen tiefen Netzes ist unter http://www.cs.toronto.edu/~hinton/adi/index.htm verfügbar. Daraus ist die, in Abbildung 5 gezeigte, Darstellung der Zahlen entnommen.

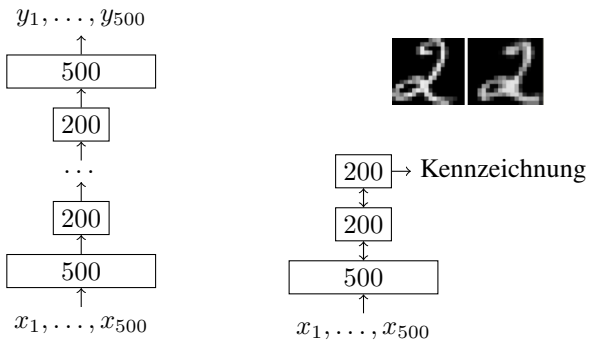

Abbildung 5. Schematische Darstellung von Autoencoder (links) und Deep Belief Net (rechts) . Beide reduzieren die Eingabe, links soll die Eingabe rekonstruiert werden, rechts wird eine Kennzeichnung (Label) zugeordnet. Oben ist die Zahl 2 (links) als Eingabe zu erkennen mit erzeugten Wahrscheinlichkeitsbereich (Fantasie, rechts), visualisierte Zwischenausgabe im Netz.

A. Spezifische Probleme

Diese Probleme sind, je nach verwendetem Netz, unterschiedlich ausgeprägt. Meistens ist bereits untersucht, wie sich diese durch Anpassung der verwendeten Struktur vermindern lassen. Dadurch tritt lediglich eine Verschlechterung der Erkennung auf, welche jedoch nach wie vor möglich ist.

1) Stabilitäts-Plastizitäts-Dilemma: Die Kapazität an unterscheidbaren Eingaben, die durch ein künstliches neuronales Netz repräsentiert werden können, sind begrenzt. Es tritt das Problem auf, dass bereits Gelerntes nicht vergessen werden soll. In [5] wird untersucht, wie durch Netzanpassungen dieses Problem abgeschwächt werden kann.

2) Overfitting: Bei trainierten Netzen können sehr gute Ergebnisse auf den Testdaten erzielt werden, jedoch mit dem Effekt, ähnliche Eingaben nicht zuordnen zu können. Das kann darauf beruhen, dass viele Merkmale zu spezifisch betrachtet werden[9] und daher eine Generalisierung nicht möglich ist. Anschaulich wird es als Auswendiglernen bezeichnet, wodurch Transferaufgaben nicht souverän gelöst werden können. Verhindert werden kann das durch mehrfaches Aufteilen der Testdaten, so dass nicht relevante Merkmale besser verteilt sind oder auch durch eine Einschränkung bei der Anzahl der Parametern und Neuronen. Nach [6] existieren auch praktische, erfolgreiche Ansätze.

B. Generelle Probleme

Solche Probleme treten unabhängig von Netztopologie und Training auf und lassen sich nicht durch geringfügige Anpassungen vermindern. Diese sorgen dafür, dass die Erkennung generell nicht wie vorgesehen funktioniert bei bestimmten Eingaben. In [7] wird das als *blinder Fleck* bezeichnet als Analogie zum menschlichen Auge. Auf den allgemeinen Fähigkeiten in III aufbauend wird untersucht, wie sich die Klassifizierung zu einem falschen Ergebnis verleiten lässt. Anhand nachfolgenden Tabelle lässt sich unterscheiden, dass meist die Erzeugung von *richtigen* Ergebnisse betrachtet wird. Es lassen

[9]In der Regressionsanalyse bedeutet das, dass zu viele erklärende Variablen vorhanden sind.

sich zwei Ansätze betrachten, welche die *falsche* Vorhersagen erzeugen wollen[10].

	x gehört zu K	x gehört nicht zu K
x ist K zugeordnet	richtig positiv	**falsch positiv**
x ist K nicht zugeordnet	**falsch negativ**	richtig negativ

1) Erzeugen von falsch negativen Klassifikationen: Eine Eingabe x gehört zu einer Klasse, wird jedoch nicht dieser zugeordnet. Nach [7] können richtig zugeordneten Eingaben verändert werden, so dass eine menschliche Zuordnung weiterhin möglich ist, jedoch die Klassifizierung mittels neuronaler Netze nicht mehr gegeben ist. Einerseits kann die ursprüngliche Eingabe durch ein zufälliges Rauschen gestört werden. Nach den experimentellen Ergebnissen in [7, S.7] sinkt die Erkennungsrate auf die Hälfte. Andererseits wird ein minimale Störung berechnet, welche die Eingabe so wenig wie möglich verändert und dennoch eine erfolgreiche Klassifizierung nahezu unmöglich macht. Das kann zum Beispiel über heuristische Verfahren ähnlich dem Gradientenabstiegsverfahren[11] erfolgen, mit dem Ziel den Fehler durch Anpassung der Eingabe, statt der Gewichte, zu maximieren.

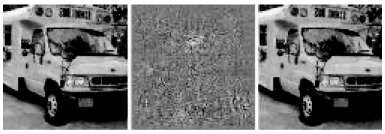

Abbildung 6. Links: Abbildung wird korrekt klassifiziert. Rechts: Abbildung nicht klassifizierbar. Mitte: Differenz. Nach [7, Abbildung 5]

2) Erzeugen von falsch positiven Klassifikationen: Es ist möglich eine Klassifikation mit 99,99% Sicherheit zu erreichen bei einer Eingabe, die von Menschen nicht klassifizierbar ist. In [8] werden unter anderem Verfahren vorgestellt, die für eine zufällige Eingabe mittels einem evolutionären Algorithmus eine sichere Klassifikation erreichen. Je nach Art der Kodierung (direkt oder indirekt) und Ausgangswerte (zufällige oder natürliche Daten) entstehen Eingaben, welche vom Rauschen bis hin zu generierten, jedoch sinnlosen, Abbildungen reichen können.

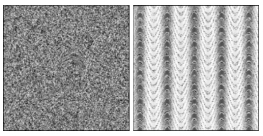

Abbildung 7. Links: indirekte, rechts: direkte Kodierung, welche jeweils als Pfau klassifiziert wird. Nach [8, Abbildung 1]

C. Bedeutung der Ergebnisse

Mit den Ergebnissen, welche die bisherigen Verfahren auf Robustheit testen, werden strukturelle Schwächen aufgezeigt. Die Ergebnisse in [7, S.6] und praktischen Ausführungen in [8, Anhang A] zeigen, dass ein Großteil der erzeugten Eingabemuster auch bei anderen Netztopologien und Einstellungen zu Fehlern führen und das es außerdem unerheblich ist, auf welchen Eingabedaten diese trainiert werden. Somit kann das nicht nur durch spezifische Probleme erklärt werden.

1) Begründung: Bei tiefen Netzen kann nicht explizit gesteuert werden, welche Merkmale abstrahiert werden sollen. Daraus wird in [8, S.5] gefolgert, dass die Merkmale zu speziell sind und nicht die globale Struktur erschlossen wird. Nach [7, S.4] führt die Betrachtung der Umgebung der eigentlichen Eingabe zu der Möglichkeit, dass sich ein neuronales Netz durch Rauschen stören lässt.

2) Eigene Folgerungen: Durch wenige Änderungen der Eingaben (Abschnitt IV-B1) oder durch eine willkürliche Eingabe (Abschnitt IV-B2) kann die Klassifizierung gestört werden. Die Verfahren dafür nutzen aus, dass in einigen Eingabebereichen Schwankungen zu signifikanten Unterschieden bei der Bewertung führen können. Beim Gradientenabstieg entspricht so eine Stelle einer starken Steigung oder bei stochastischer Betrachtung dem Aufbrechen der Korrelation durch einzelne signifikante Ausreißer.

V. Zusammenfassung und Ausblick

Die Entwicklung hat durch den Backpropagation-Algorithmus zunächst große Fortschritte gemacht, welcher jedoch nicht die Erwartungen erfüllt hat. Mit dem Ansatz Deep Learning wurden nach vielversprechenden Ergebnissen wiederum strukturelle Probleme entdeckt. Diese sind bisher nicht in der Praxis aufgetreten, da dieser blinder Fleck sehr spezielle Eingabemuster erfordert. Es zeigt sich jedoch, dass die Steuerung der Merkmalsextraktion zur Erkennung globaler Zusammenhänge weiter erforscht werden muss. Insbesondere im Hinblick auf den Einsatz der Technologie in kritischen Bereichen, wie autonome Fahrzeugsteuerung, dürfen nachweislich Probleme mit einer hohen, dennoch falschen Sicherheit nicht auftreten.

[10] Es sei darauf hingewiesen, dass die Verifikation (x gehört zu) vom Menschen getroffen wird.

[11] [7, S.5] erwähnt eine Approximation aus der Klasse der Quasi-Newton-Verfahren, welche auf dem selben Prinzip beruht.

Literatur

[1] D. Kriesel, *Ein kleiner Überblick über Neuronale Netze*. http://www.dkriesel.com, 2007.

[2] H. Lee *et al.*, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," 2009, http://www.cs.toronto.edu/~rgrosse/icml09-cdbn.pdf.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," Apr 2014, http://arxiv.org/abs/1206.5538.

[4] L. Deng, "Three classes of deep learning architectures and their applications : A tutorial survey," 2013, http://research.microsoft.com/pubs/192937/Transactions-APSIPA.pdf.

[5] R. Coop and I. Arel, "Mitigation of catastrophic forgetting in recurrent neural networks using a fixed expansion layer," 2013, http://web.eecs.utk.edu/~itamar/Papers/IJCNN2013.pdf.

[6] N. Srivastava *et al.*, "Dropout: A simple way to prevent neural networks from overfitting," 2014, http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf.

[7] C. Szegedy *et al.*, "Intriguing properties of neural networks," Feb 2014, http://arxiv.org/abs/1312.6199.

[8] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," Dez 2014, http://arxiv.org/abs/1412.1897.