
Wozu wird Deep Learning verwendet?

We will have intelligent machines. It's clearly a matter of time. We will have machines that, without being very smart, will do useful things, like drive our cars autonomously.

Yann LeCun

Das Gewinnen von Wettbewerben im Bereich des Machine Learning läutete die Renaissance der künstlichen neuronalen Netze ein. So gewannen die ersten rekurrenten neuronalen Netze (RNNs) von Jürgen Schmidhuber et al. ab 2009 eine Reihe international anerkannter Mustererkennungswettbewerbe. In den darauffolgenden Jahren gewannen ähnliche Ansätze weitere relevante Wettbewerbe. Seit dieser Zeit ist viel passiert. Firmen wie Google, Facebook, Apple, Microsoft, IBM, Amazon und andere integrieren die neuesten Deep-Learning-Erkenntnisse in ihre eigenen Produkte oder stellen sie als *Software as a Service* (SaaS) zur Verfügung. Neben den reinen Diensten bieten auch immer mehr Firmen eigene Programmierbibliotheken an oder unterstützen bestehende mit Geld, Personen oder Cloud-Rechenzeit. Dazu kommt der »Krieg um die Köpfe«. Absolventen bekannter Machine-Learning-Fakultäten bleiben selten an den Universitäten, sondern wechseln direkt nach dem Studium zu einem der großen Player im Bereich Deep Learning.

Im Folgenden sollen Produkte und Dienste vorgestellt werden, die in den letzten Jahren mithilfe von Deep Learning entstanden sind. Besonders in den Anwendungsfeldern der maschinellen Übersetzung und der Bilderkennung kann man den massiven Einfluss von Deep Learning erkennen, der zunehmend auch auf unseren Computern und Mobiltelefonen Anwendungen verbessert.

Die Vielfalt der Anwendungen im folgenden Überblick leitet sich dabei nicht von der Vielzahl unterschiedlichster Algorithmen ab, sondern basiert auf wenigen Frameworks, die eine für die Anwendung spezifische Netzwerkarchitektur implementieren, die im Kern aber immer aus künstlichen neuronalen Netzen bestehen. Dies ist einer der grundlegenden Unterschiede zu klassischen Lösungen der Informatik im Allgemeinen und zum Machine Learning im Speziellen.

Deep Learning bei Google

Google gründete 2011 das Projekt *Google Brain*, das zuallererst ein Forschungsprojekt der Google-Mitarbeiter Greg Corrado und Jeff Dean zusammen mit dem Professor für Computer Science an der Stanford-Universität, Andrew Ng, war.

Ziel des Projekts war es, ein Softwaresystem zu erstellen, das auf Googles Cloud-Architektur ausgeführt wurde und ein hohes Maß an Skalierung ermöglichte, um auf großen Datenmengen arbeiten zu können. Das Ergebnis der Zusammenarbeit war DistBelief, ein Programmierframework, das, auf mehrere GPUs verteilt, die Implementierung von Deep-Learning-Algorithmen ermöglichte (<https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/40565.pdf>).

Einige Google-Produkte nutzten DistBelief bereits, als im Juni 2012 weitere Details zu Googles Aktivitäten im Bereich der Entwicklung von Anwendungen auf Basis künstlicher neuronaler Netze in einem Artikel in der New York Times veröffentlicht wurden (<http://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html>).

Mit dem Kauf der in Toronto, Kanada, ansässigen Firma DNNresearch 2013 verstärkte Google sein Brain-Team um Alex Krizhevsky, Ilya Sutskever und Geoffrey Hinton (https://www.wired.com/2013/03/google_hinton/).

Im Januar 2014 kaufte Google schließlich die englische Deep-Learning-Firma DeepMind, die vier Jahre zuvor von Demis Hassabis, Shane Legg und Mustafa Suleyman gegründet wurde. Ende 2013 hatte DeepMind seine Arbeit zum Thema bestärkendes Lernen (engl. Reinforcement Learning) vorgestellt, bei dem sich ihr Modell selbstständig das Spielen klassischer Atari-2600-Videospiele beigebracht hatte.

TensorFlow

2015 veröffentlichte Google mit *TensorFlow* die Weiterentwicklung von DistBelief als Open-Source-Framework unter der Apache-2.0-Lizenz. Einen passenden Spielplatz (<http://playground.tensorflow.org/>) eröffnete Google 2016 (Abbildung 3-1). Im Februar 2017 kam die Version 1.0 von TensorFlow heraus. Nur einen Monat später wurde das *Sonnet* (<https://github.com/deepmind/sonnet>) genannte Framework von Googles Tochter DeepMind als Open Source veröffentlicht. Hintergrund der Veröffentlichung war, dass DeepMind komplett auf TensorFlow als führendes Framework umgestiegen ist. Sonnet bietet einen objektorientierten Ansatz bei der Definition künstlicher neuronaler Netze, ähnlich wie es das Neural-Network-Paket (<https://github.com/torch/nm>) von Torch (<http://torch.ch>) ermöglicht. Ende August 2017 folgte die TensorFlow-Version 1.3. TensorFlow ist der große Stern am Himmel der Deep-Learning-Frameworks. Das belegen nicht nur die Zahlen des von Google betriebenen GitHub-Repositories: über 28.000 Forks und über 800 Contributors (<https://github.com/tensorflow/tensorflow>).

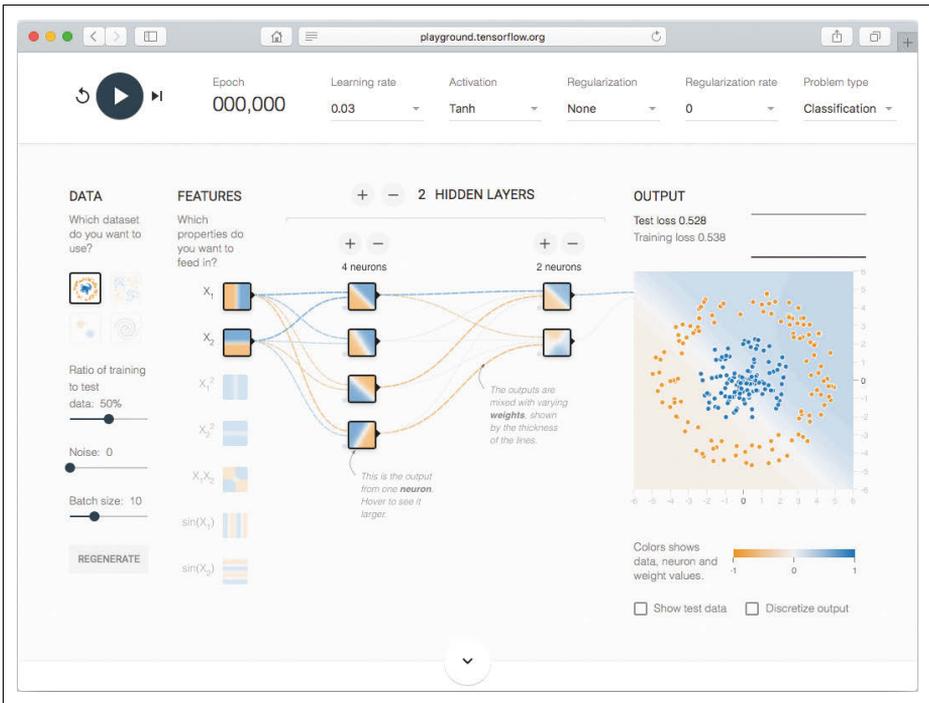


Abbildung 3-1: Mithilfe des TensorFlow-Playground lässt sich die Funktionsweise künstlicher neuronaler Netze interaktiv ausprobieren.

Seit 2016 stellt Google kontinuierlich neue Tools und Produkte basierend auf Machine Learning-Algorithmen vor, die auf TensorFlow basieren.

AlphaGo

Die wohl publikumswirksamste Aktion, die Googles Deep-Learning-Aktivitäten der breiten Masse zugänglich machte, war der Gewinn einer Go-Partie gegen den professionellen Dan-9-Spieler Lee Sedol. Die fünf Matches, die vom 9. bis 15. März 2016 im südkoreanischen Seoul gespielt und als Livestream übertragen wurden, endeten 4 : 1 für die AlphaGo-Anwendung von Googles DeepMind (<https://www.youtube.com/watch?v=vFr3K2DORc&list=PLqYmG7hTraZA7v9Hpbps0QNmJC4L1NE3S>).

Ende Mai 2017 trat AlphaGo ein zweites Mal an, sich mit der Weltspitze der Go-Spieler zu messen. Im Rahmen des Future of Go Summit (<https://events.google.com/alphago2017/>) im chinesischen Wuzhen forderte AlphaGo nicht nur den Weltranglistenersten, den Chinesen Ke Jie, heraus, sondern versuchte auch, gegen ein Team aus fünf der weltbesten Go-Spieler zu gewinnen. Die Ergebnisse waren eindeutig: AlphaGo besiegte sowohl Ke Jie in drei Spielen als auch das Team. AlphaGo bekam dafür von der Chinese Go Association den 9. Dan verliehen.

Im Gegensatz zur AlphaGo-Version von 2016 kommt die aktuelle Version mit einem Zehntel der Rechenpower aus und wurde auf einem einfachen Google-Cloud-System mit *Tensor Processing Units* (TPUs) ausgeführt.

Google Photo

Ein weiteres Produkt mit Deep-Learning-Hintergrund brachte Google im Oktober 2016 mit *Google Photo* heraus. Die Anwendung, die sowohl im Web als auch mobil als App kostenlos nutzbar ist, speichert Bilder auf Google Cloud. Googles Deep-Learning-Algorithmen helfen dabei, die eingehenden Bilder direkt zu verschlagworten, um nach beliebigen Begriffen innerhalb der Fotos einfach suchen zu können. Abbildung 3-2 zeigt die Fotosuche auf einem iPhone. Der Begriff *Kaffeetasse* ist nicht innerhalb der Bildinformation verfügbar, beim Hochladen der Bilder wurden diese von Google Photo selbstständig mit Begriffen versehen. Über das Suchfeld können entsprechende Fotos zu den Begriffen gefunden werden.

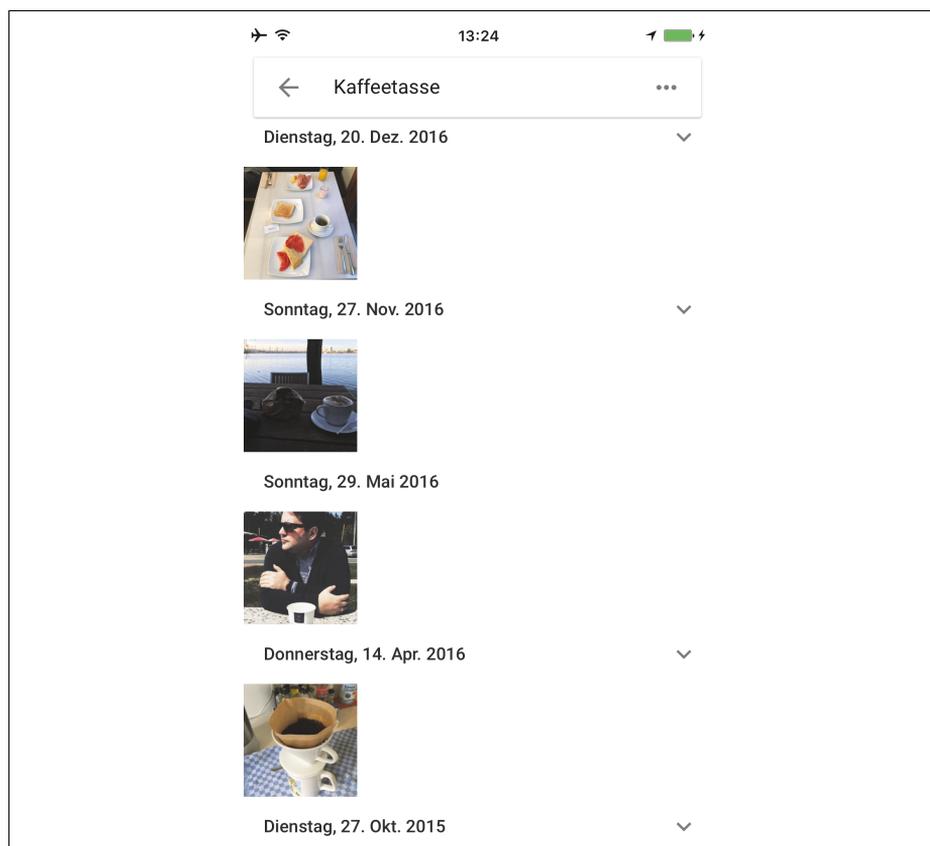


Abbildung 3-2: Suche innerhalb der Google-Photo-App für iOS nach dem Schlagwort *Kaffeetasse*. Die Ergebnisse wurden automatisch von Google verschlagwortet.

Google Translate

Ende 2016 wurde *Google Translate* auf ein künstliches neuronales Netz umgestellt (<https://research.googleblog.com/2016/11/zero-shot-translation-with-googles.html>). Neu ist, dass dieses Modell, Google Neural Machine Translation (GNMT, <https://research.google.com/pubs/pub45610.html>), alle 103 Sprachen übersetzen kann und nicht mehr ein Modell pro Sprache nutzt. Der große Vorteil dieser multilingualen Methode ist, dass jetzt auch Beispiele aus Sprachen mit hohem Verbreitungsgrad für Sprachen mit wenigen Daten genutzt werden können. GNMT verwendet sogenannte rekurrente neuronale Netze (RNNs), die mit Schichten aus Long-Short-Term-Memory-Neuronen (LSTMs) aufgebaut werden (Näheres zu RNNs und LSTMs finden Sie in Kapitel 6, *Deep-Learning-Anwendungen*). Eine genaue Beschreibung der Methode findet sich auf arXiv.org (<https://arxiv.org/abs/1609.08144>). Zusätzlich zur Webseite von Google Translate können auch die mobilen Apps zur Sprachübersetzung genutzt werden. Neben der eigentlichen Übersetzung nach einer Texteingabe nutzen die mobilen Apps auch das Mikrofon, die Kamera und das Touch-Display für die Text- bzw. Spracheingabe.

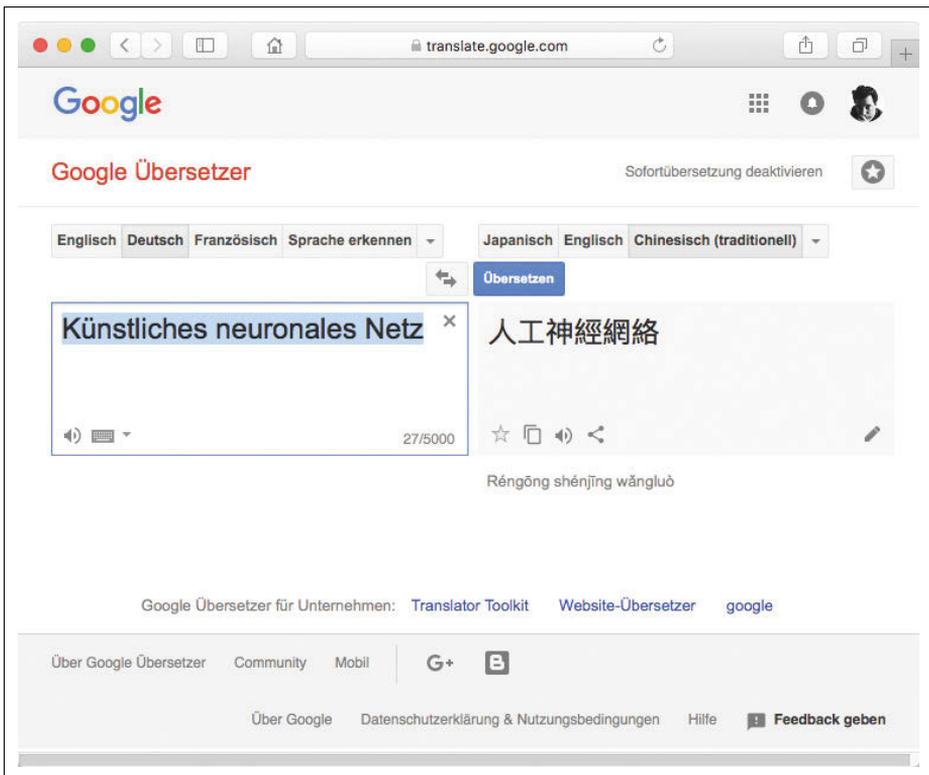


Abbildung 3-3: Auf der Webseite translate.google.com lassen sich Texte mit bis 5.000 Zeichen in über 100 Sprachen übersetzen.

Google Cloud ML

Mitte 2016 brachte Google auf der Google-Cloud-Plattform den Dienst *Google Cloud Machine Learning* heraus. Der Cloud-Dienst beinhaltet dabei Tools für die Bild- und Videoanalyse sowie die Text- und Spracherkennung mit deren Übersetzung. Alle Dienste lassen sich über APIs für die Programmiersprachen Python, Java, Node.js, PHP, .NET, Go und Ruby ansprechen. Die Preise für die Nutzung der Dienste richten sich nach der Menge der Aufrufe. Je öfter man die Dienste nutzt, desto günstiger wird der Preis pro Aufruf. So kostet zum Beispiel die Erkennung von Bildinhalten über die Cloud-Vision-API bei einer Million Aufrufen jeden Monat rund 1,50 Dollar pro zurückgeliefertes Label. Ab 5 Millionen Aufrufen pro Monat sinkt der Preis auf 1,00 Dollar. In Zukunft werden TensorFlow und Cloud ML noch stärker zusammenwachsen. In Kapitel 7, *Deep Learning und Big Data*, und Kapitel 8, *Deep Learning produktiv*, werde ich noch einmal auf Google-Cloud-ML-Angebote zurückkommen, um zu zeigen, wie sich eigene Deep-Learning-Anwendungen innerhalb der Cloud nutzen lassen.

Google AutoDraw

Eine simple, aber effektive Methode, um handschriftliche Zeichnungen in professionelle Figuren zu verwandeln, bietet seit April 2017 die Google-Software *AutoDraw*. Auf einer Malfläche lässt sich eine Freihandzeichnung anfertigen. In der oberen Menüleiste werden dazu passende Symbole angezeigt, die AutoDraw aus den bisherigen Strichen erkennt. Abbildung 3-4 zeigt auf der linken Seite die handgemalten Konturen einer Tasse. Nach Auswahl des entsprechenden Icons aus der Menüleiste ändert sich die Abbildung zu der Kaffeetasse auf der rechten Seite von Abbildung 3-4. AutoDraw kann auf dem Desktop und auf mobilen Endgeräten genutzt werden.

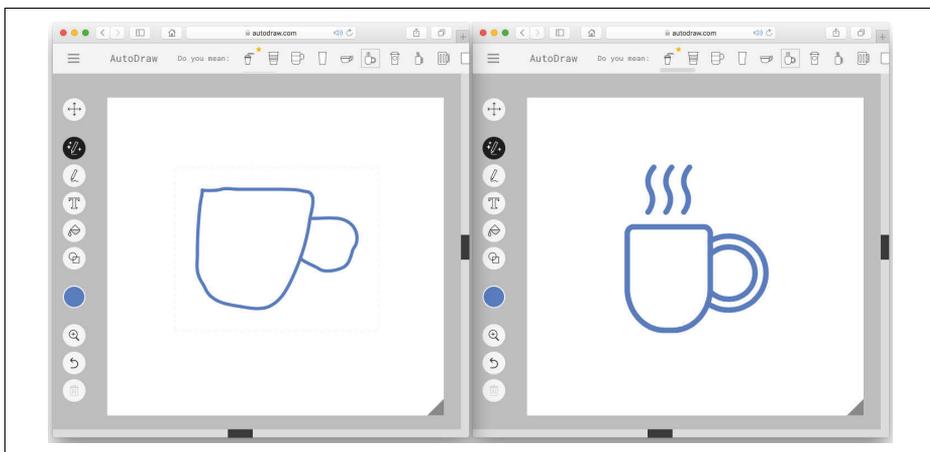


Abbildung 3-4: Google AutoDraw schlägt automatisch Formen für die eigenen Freihandzeichnungen vor.

Google self-driving car

Ein weiteres Projekt, das sich Deep-Learning-Algorithmen zunutze macht, ist *Google self-driving car*. Seit Ende 2016 hat die Google-Mutterfirma Alphabet die Technologie von *self-driving car* an ihre Tochter Waymo ausgelagert und verspricht als Ziel des Projekts nichts Geringeres als das fahrerlose Auto. Google arbeitet an diesem Projekt bereits seit über sechs Jahren und reichte erstmals 2011 ein Patent für die Technik und den Betrieb fahrerloser Autos ein. Deep-Learning-Technologien werden vor allem dort eingesetzt, wo eine Fülle von Sensordaten erkannt, klassifiziert und für die weitere Steuerungsentscheidung aufbereitet werden muss. Dies gilt insbesondere für die Erkennung der Straße, von anderen Autos, Straßenschildern, Ampeln und Hindernissen.

Die hier genannten Beispiele können nur an der Oberfläche kratzen und basieren auf öffentlich zugänglichen Informationen. Wie stark Google an weiteren Anwendungen und Produkten im Kontext von Deep Learning arbeitet und welche Rolle dabei TensorFlow und Googles Cloud ML spielt, wird die Zukunft zeigen.

Deep Learning bei Facebook

Im Dezember 2013 verpflichtete Facebook den französischen Deep-Learning-Innovator Yann LeCun für das firmeneigene Forschungslabor für künstliche Intelligenz.

Facebook nutzt Deep-Learning-Technologien, um Posts zu übersetzen, Fotos zu klassifizieren und die richtige Werbung an die richtigen Nutzer auszuliefern.

Informationen darüber, welche Deep-Learning-Tools und -Anwendungen Facebook intern nutzt, kommen nur vereinzelt an die Öffentlichkeit. Bekannt ist, dass Facebook mit *FBLearn Flow* und *FBLearn Predictor* zwei selbst entwickelte Tools für die Implementierung von Machine-Learning- und Deep-Learning-Anwendungen im eigenen Backend nutzt. In einem Blogeintrag vom Mai 2016 beschreibt Jeffrey Dunn, seines Zeichens Softwareentwickler bei Facebook, die einzelnen Backend-Komponenten (<https://code.facebook.com/posts/1072626246134461/introducing-fblearn-flow-facebook-s-ai-backbone/>).

FBLearn Flow dient dabei als Rückgrat von Facebooks Entwicklungen rund um Machine Learning. Innerhalb von FBLearn Flow können Workflows über eine Nutzerschnittstelle definiert werden. Abbildung 3-5 zeigt das von Facebook intern verwendete System. Ähnlich wie Googles TensorBoard zeigt FBLearn sowohl den Berechnungsgraphen als auch weitere Ansichten an.

Basierend auf FBLearn Flow, wird die sogenannte Lumos-Plattform genutzt, um Bilder und Videos zu analysieren. Lumos ermöglicht es, Fotos textuell zu beschreiben.

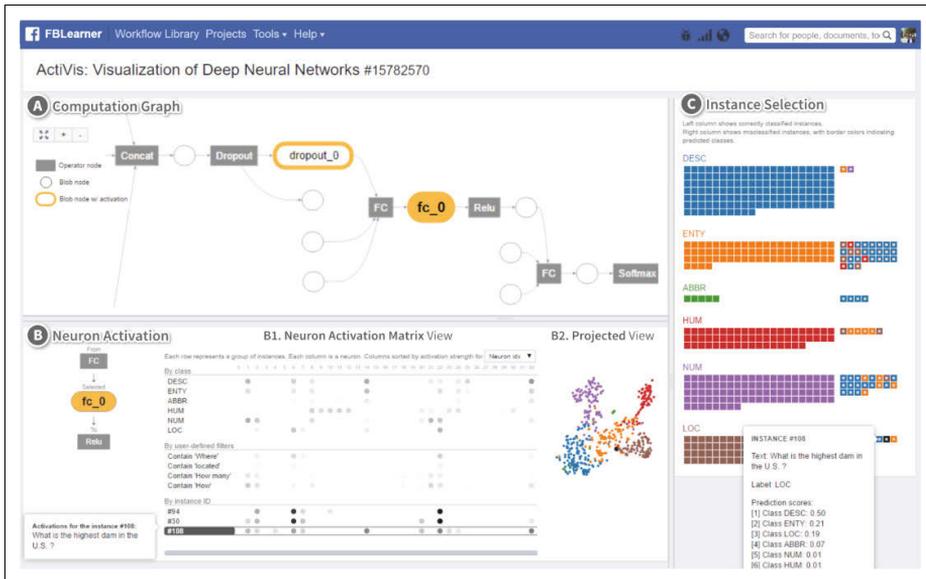


Abbildung 3-5: Das von Facebook für interne Zwecke entwickelte System FBLeaRner Flow

Darüber hinaus sind zwei weitere Anwendungen bekannt, die Facebook zur Textanalyse und Gesichtserkennung von Personen einsetzt.

- DeepText (<https://code.facebook.com/posts/181565595577955/introducing-deeptext-facebook-s-text-understanding-engine/>) ermöglicht die Textanalyse von mehr als 20 Sprachen innerhalb einer verteilten Big-Data-Architektur. DeepText basiert dabei auf Convolutional Neural Networks (CNNs), um große Textmengen sowohl zu klassifizieren als auch eine Stimmungsanalyse (Sentiment Analysis) durchzuführen. Die Grundlagen von DeepText wurden von Yann LeCun und Xiang Zhang im Paper *Text Understanding from Scratch* auf arXiv.org veröffentlicht (<https://arxiv.org/abs/1502.01710>).
- DeepFace (<https://research.fb.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/>) ermöglicht die Erkennung von Personen auf Fotos. Diese Funktionalität darf von Facebook innerhalb der EU nicht genutzt werden.

Darüber hinaus entwickeln die Mitarbeiter von Facebook AI Research (FAIR, <https://research.fb.com/category/facebook-ai-research-fair/>) an etlichen Modulen für die Machine-Learning- und Deep-Learning-Frameworks Torch und PyTorch mit. Folgende Teilprojekte gehen dabei auf das Konto von FAIR:

- iTorch (<https://github.com/facebook/iTorch>) – Ist ein IPython-Kernel für die Verwendung von Torch innerhalb von Jupyter Notebooks. Es ermöglicht die Autovervollständigung der Torch-Funktionen und die Einbettung von Bild- und Audioinformationen.

- `fbcunn` (<https://github.com/facebook/fbcunn>) – Dabei handelt es sich um eine Reihe von Erweiterungen für das Torch-Framework. Dazu gehören unter anderem die Implementierung eines hierarchischen Softmax, ein Container zur parallelen Ausführung von Modellen über mehrere GPUs sowie eine beschleunigte Implementierung der CuNN.
- `fbnn` (<https://github.com/facebook/fbnn>) – Sind Erweiterungen für das Neuronale-Netzwerk-Modul von Torch.
- `fblualib` (<https://github.com/facebook/fblualib>) – Enthält Erweiterungen und Tools für die Nutzung von Lua und Torch. Dazu gehören unter anderem eine Bibliothek zum Einlesen von Matlab-Dateien, eine auf Thrift basierte Klasse für die schnelle Objektserialisierung und -deserialisierung und vieles mehr.

Hier ist Facebook sehr freigiebig in der Rückgabe der eigenen Entwicklungen an die Community.

Seit der Entwicklungskonferenz F8 im April 2017 ist bekannt, dass Facebook mit `Caffe2` das bekannte `Caffe` unter Open-Source-Lizenz weiterentwickelt. Yangqing Jia, der `Caffe` an der Universität von Kalifornien in Berkeley entwickelt hat, war im Februar 2016 zu Facebook gegangen, um seine Entwicklungen fortzuführen. Zuvor war Yangqing Jia zwei Jahre bei Google Research unter Vertrag, nachdem er 2013 in Berkeley seinen Abschluss machte und `Caffe` entwickelte.

Deep Learning bei IBM

Eine der wichtigsten Marken von IBM, wenn es um die Demonstration von künstlicher Intelligenz geht, ist das Computerprogramm `Watson` (<https://www.ibm.com/watson/>). Bereits 2011 zeigte IBM die Fähigkeiten von `Watson` im Rahmen einer Folge der Quizsendung `Jeopardy!`. Hier musste `Watson` passende Fragen zu vorgegebenen Antworten formulieren. `Watson` trat dabei gegen zwei menschliche Gegner an, die in vorherigen Folgen von `Jeopardy!` als Gewinner hervorgegangen waren. `Watsons` Stärke ist die Verarbeitung semantischer Informationen und die Generierung von Hypothesen. Ebenso wie Google Cloud Platform bietet IBM die `Watson-API` über die hausinterne Cloud-Lösung `Bluemix` an.

Neben `Watson` stellt IBM mit `PowerAI` eine Plattform für verteilte Deep-Learning-Anwendungen basierend auf `Caffe`, `Torch`, `Theano`, `Chainer` (<http://chainer.org>) und `TensorFlow` bereit. Die Anwendungen laufen dabei auf IBM-eigenen Power-Systemen. IBM versucht, mit `PowerAI` eine Distribution für Deep-Learning-Frameworks zu schaffen, die auf der IBM-eigenen Power-CPU laufen. `PowerAI` unterstützt dabei auch `NVIDIA-GPUs` für die Beschleunigung der Berechnungen im Deep-Learning-Kontext.

Deep Learning bei Microsoft

Bereits 2011 führte Microsoft künstliche neuronale Netze zur Sprachverarbeitung ein. Das erklärte Ziel von Harry Shum, Executive Vice President bei Microsoft Research, und Xuedong Huang, Chief Speech Scientist, ist es, einen Universalübersetzer zu schaffen, der jede Sprache in jede andere Sprache übersetzen kann – und das im Idealfall in Echtzeit (<http://news.microsoft.com/features/speak-hear-talk-the-long-quest-for-technology-that-understands-speech-as-well-as-a-human/>). Die Ergebnisse dieser Forschungsarbeit präsentierte Microsoft 2015 mit dem Produkt Skype Translator (<https://www.skype.com/en/features/skype-translator/>), einem Skype-Plugin, das Sprach- und Videotelefonate in den Sprachen Chinesisch (Mandarin), Englisch, Französisch, Deutsch, Italienisch, Portugiesisch, Spanisch, Russisch und Arabisch simultan übersetzen kann. Weitere 60 Sprachen können bei rein textuellen Nachrichten übersetzt werden (Abbildung 3-6).

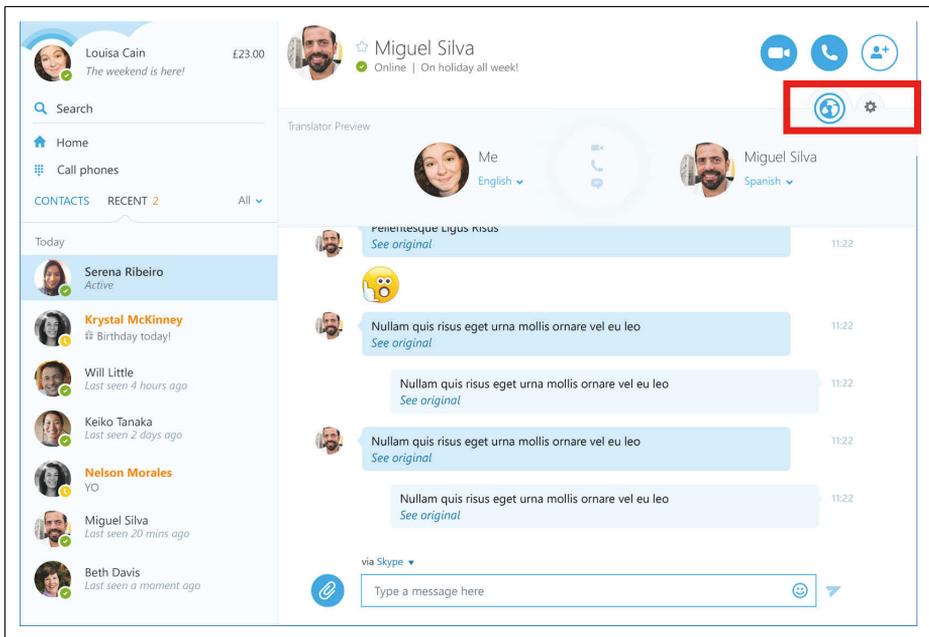


Abbildung 3-6: Skype Translator in Aktion

Im Januar 2016 stellte Microsoft mit dem *Computational Network Toolkit* (CNTK) auf GitHub (<https://github.com/Microsoft/CNTK>) ein Open-Source-Framework für die Implementierung von Deep-Learning-Anwendungen zur Verfügung. Dabei läuft CNTK nicht nur auf Windows-Systemen, sondern kann auch unter Linux installiert werden. Wie viele andere Frameworks verfügt CNTK sowohl über eine C++- als auch über eine Python-Schnittstelle.

Deep Learning bei Baidu

Baidu ist das Google Chinas. Als größter Suchmaschinenanbieter im asiatischen Markt belegt Baidu mittlerweile Platz 4 der reichweitenstärksten Webseiten der Welt hinter Google, YouTube und Facebook (Quelle: The top 500 sites on the web, <http://www.alexa.com/topsites>).

Ähnlich wie Google mit TensorFlow hat Baidu in 2016 das Deep-Learning-Framework *PaddlePaddle* (<http://www.paddlepaddle.org>) veröffentlicht, das ein breites Anwendungsspektrum abdecken soll.

PaddlePaddle ist in C++ implementiert und im Quellcode über GitHub (<https://github.com/PaddlePaddle/Paddle>) verfügbar. Die Dokumentation gibt es sowohl in Englisch als auch in Chinesisch.

Ebenso wie Caffe und TensorFlow ermöglicht PaddlePaddle das Training auf einer verteilten Plattform mit mehreren Rechnern innerhalb eines Clusters oder auf mehreren Grafikkarten. Ähnlich der Caffe-Integration mit Spark (siehe Kapitel 7, *Deep Learning und Big Data*) nutzt auch PaddlePaddle dieses Framework in Kombination mit dem Ressourcenmanager YARN von Hadoop, um über mehrere Systeme Modelle zu trainieren.

Viele Machine-Learning-Themen werden bei Baidu Research (<http://research.baidu.com>) in Sunnyvale, Kalifornien, entwickelt. Dort wurde, in Zusammenarbeit mit NVIDIA, 2015 ebenfalls das GPU-unterstützte *Deep Speech* vorgestellt, das Spracherkennung mithilfe von Tesla-Grafikkarten ermöglicht – und das sowohl für Englisch als auch für Mandarin.



Abbildung 3-7: Ein Foto von mir mit ungewöhnlicher Fellmütze fördert zumindest drei ähnliche Bilder aus Baidus Bildersuchmaschine zutage.

Neben der Spracherkennung hat Baidu 2017 einen Sprachassistenten mit Namen DuerOS im Angebot. In China ist er bereits in HTC's U11-Smartphone integriert.

Ein zweiter Anwendungsschwerpunkt für Deep Learning liegt bei Baidu auf der Bilderkennung. Innerhalb der Suchmaschine gibt es einen Dienst, der das Hochladen eines Fotos ermöglicht, um ähnliche Bilder im Web zu suchen (Abbildung 3-7).

Auch für selbstfahrende Autos à la Google hat Baidu eine Lösung in Vorbereitung. Das Apollo-Projekt (<http://apollo.auto/index.html>), von Baidu als offene Plattform für autonomes Fahren konzipiert, soll es Entwicklern ermöglichen, eine eigene Software auf dessen Grundlage zu erstellen. Auf GitHub (<https://github.com/apolloauto>) ist bereits Version 1.0 der Plattform verfügbar.

Deep Learning bei Apple

Apple nutzt Deep Learning innerhalb einiger Anwendungen, wie zum Beispiel der macOS-Foto-App für das Auffinden von Personen und Gesichtern (Abbildung 3-8) oder des Sprachassistenten Siri. Apple verfolgte hier bisher allerdings eine andere Strategie als Google oder Facebook: Statt viele der Eigenentwicklungen in ein Framework zu integrieren und dieses als Open Source zur Verfügung zu stellen, hielt man sich bei Apple lange Zeit eher bedeckt, was Machine Learning anging. Bis zur World Wide Developers Conference 2017 waren lediglich Frameworks wie das *Basic Neural Network Subroutines* (BNNS) sowie *Metal Performance Shaders* für die Berechnung innerhalb von GPUs für Entwickler verfügbar.

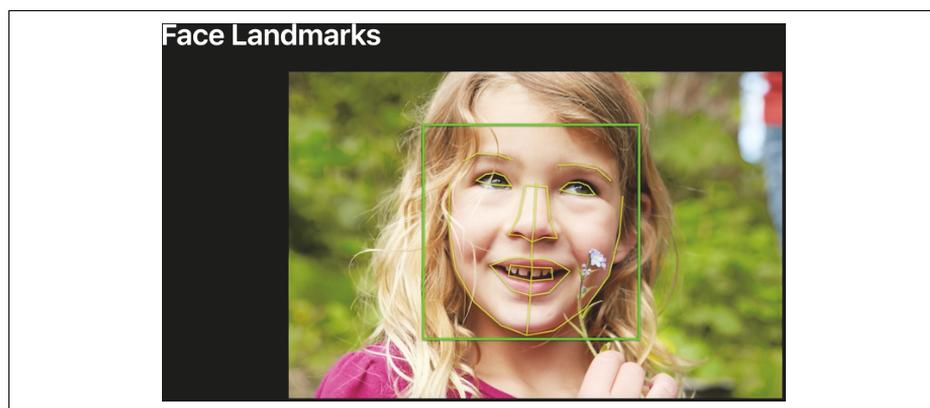


Abbildung 3-8: Apples Vision-Framework, das auf Core ML aufbaut, erkennt auch Gesichtsmarkierungen und deren Positionen (Quelle: <https://developer.apple.com/videos/play/wwdc2017/506/>).

Im Juni 2017 stellte Apple dann mit dem Framework Core ML eine Bibliothek vor, die eine einheitliche Schnittstelle für macOS, iOS, watchOS und tvOS realisiert.

Core ML implementiert dabei Funktionen, die von High-Level-Frameworks wie Vision (<https://developer.apple.com/documentation/vision>), NLP (<https://developer.apple.com/videos/play/wwdc2017/208/>) und GameplayKit (<https://developer.apple.com/documentation/gameplaykit>) genutzt werden können.

Besonders im mobilen Umfeld ermöglicht es Apple damit, eine standardisierte Schnittstelle für alle Apps zu schaffen, die sowohl Deep-Learning- als auch Machine-Learning-Algorithmen nutzen wollen.

Um bereits existierende Modelle im Kontext von Core ML weiter nutzen zu können, steht mit den freien Core ML Tools (<https://pypi.python.org/pypi/coremltools>) eine Python-Bibliothek zur Verfügung, die bestehende Deep-Learning-Modelle von Caffe und Keras in Core-ML-Modelle überführen kann.

Eine weitere Neuheit wurde mit dem Blog »Apple Machine Learning Journal« (<https://machinelearning.apple.com>) gestartet. Seit Juli 2017 erscheinen dort in unregelmäßigen Abständen wissenschaftliche Artikel rund um das Thema Machine Learning und Deep Learning.

Deep Learning bei Amazon

Amazon betätigt sich schon seit einigen Jahren mit verschiedensten Aktivitäten und Produkten im Bereich Deep Learning. Eines der jüngsten und sicherlich bekanntesten Produkte ist der Sprachassistent Alexa, der in den Produkten *Echo*, *Echo Dot*, *Echo Show* und weiteren Eingang gefunden hat. Ähnlich wie Apples Siri, Google Now oder Microsofts Cortana lassen sich Amazons Echo-Produkte über die natürliche Sprache steuern.

Neben Consumerprodukten arbeitet Amazon auch an weiteren Systemen rund um das Thema Deep Learning.

So veröffentlichte Amazon im Frühjahr 2016 die quelloffene C++-Deep-Learning-Bibliothek DSSTNE – ausgesprochen Destiny – auf GitHub unter Apache-Lizenz (<https://github.com/amzn/amazon-dsstne>). DSSTNE steht für *Deep Scalable Sparse Tensor Network Engine* und ermöglicht die Erstellung von Deep-Learning-Anwendungen. Amazon nutzt DSSTNE beispielsweise für die interne Erzeugung von Produktvorschlägen (engl. Recommendations) für ihre Kunden.

Anfang 2017 erklärte Amazon, dass die von ihnen unterstützte MXNet-Bibliothek (<http://mxnet.io>) Teil der Apache Foundation geworden sei. Bereits Ende 2016 verkündete Werner Vogels, CTO von Amazon.com, MXNet als Framework der Wahl, wenn es um Deep Learning im Kontext von Amazons Web Services (AWS) geht (<http://www.allthingsdistributed.com/2016/11/mxnet-defaultframework-deep-learning-aws.html>). Als die drei wichtigsten Auswahlkriterien für MXNet nannte er:

- *Skalierung* – MXNet kann ebenso wie TensorFlow und andere Frameworks nicht nur eine GPU für die Berechnung nutzen, sondern auch mehrere, die auf mehreren Rechnern verteilt sein können.
- *Entwicklungsgeschwindigkeit* – MXNet unterstützt eine breite Palette an Programmiersprachen für die Entwicklung von Anwendungen. Dazu gehören Python, C++, R, Scala, Julia, Matlab und JavaScript.
- *Portabilität* – MXNet ist auf vielen unterschiedlichen Zielsystemen lauffähig. Die in C++ implementierte Kernbibliothek mit allen Abhängigkeiten liegt in einer einzigen Quelldatei und kann auf einem Raspberry Pi 3, einem Windows-PC oder einer Amazon-Web-Services-Instanz installiert werden.

MXNet wurde maßgeblich an der Universität von Washington und der Carnegie Mellon University in Pittsburgh entwickelt. Amazon entschied sich für MXNet aufgrund seiner guten Skalierbarkeit und Performance, die in einigen Benchmarks deutlich vor Caffe und TensorFlow lagen (<https://github.com/dmlc/mxnet/issues/378#issuecomment-156730363>). Aktuell ermöglicht MXNet die Implementierung von CNNs und LSTMs.

Amazon Web Services

Im Kontext der Amazon Web Services bietet Amazon eine Reihe weiterer Dienste an, die von Deep-Learning-Algorithmen angetrieben werden.

Für die Umsetzung eigener Deep-Learning-Anwendungen stellt Amazon mit dem Deep Learning AMI (<https://aws.amazon.com/marketplace/pp/B01M0AXXQB>) eine vorkonfigurierte virtuelle Maschine bereit, die sich über Amazons Elastic-Cloud-Umgebung (EC2) mit wenigen Klicks installieren lässt und neben MXNet eine Reihe weiterer bekannter Deep-Learning-Frameworks wie Torch, Keras und Theano und natürlich auch Caffe, Caffe2 und TensorFlow enthält.

Das Deep Learning AMI ist dabei so konfiguriert, dass sich die einzelnen Tools und Frameworks bequem über ein Jupyter Notebook erreichen lassen, ganz so wie es der Docker-Container zu diesem Buch ermöglicht.

Um das AMI zu nutzen, stehen unterschiedliche Serverkonfigurationen zur Verfügung. Abhängig von Hauptspeicher, Netzwerkanbindung und Anzahl CPU- und GPU-Kerne unterscheidet sich der Preis für die Nutzung des AMI erheblich. Dazu kommt ein geringer Aufpreis für die Nutzung des AMI selbst.

Mithilfe des Deep Learning AMI lassen sich auch ganze Rechnercluster mit wenigen Klicks, verteilt in der Amazon Cloud, betreiben. Grundlage dafür ist Amazons CloudFormation, mit deren Hilfe sich fertige Clusterkonfigurationen mit einer einfachen JSON-Datei in der AWS ausrollen lassen. Für ein Deep-Learning-Cluster stehen verschiedene Konfigurationen auf GitHub bereit (<https://github.com/awslabs/deeplearning-cfn>).

Polly

Im englischen Sprachraum wird der Name Polly oft als Synonym für den Namen eines Papageis verwendet. Als Amazon Web Service ermöglicht Polly als Papagei die Wandlung von geschriebenem Text in Sprache. Polly bietet für mehr als zwei Dutzend Sprachen sowohl weibliche als auch männliche Aussprachen an. Für die Ausgabe von deutscher Sprache kann man aktuell zwischen den Stimmfarben Vicki, Marlene und Hans auswählen.

Unterstützt wird neben einfachen Texten auch die Speech Synthesis Markup Language SSML (<https://www.w3.org/TR/speech-synthesis11/>). SSML wurde vom W3C bereits 2010 als Standard der textuellen Beschreibung für die Sprachsynthese vorgestellt und wird von Firmen wie Microsoft, IBM, HP und weiteren unterstützt. Daneben bietet SSML auch die Möglichkeit, verschiedensprachige Wörter in einem einzigen Satz auszugeben. So lassen sich zum Beispiel englische oder französische Namen innerhalb eines deutschen Satzes ausgeben. Abbildung 3-9 zeigt einen mit SSML ausgezeichneten deutschen Satz, der die englischen Wörter mithilfe `<lang>`-Tags für die englische Aussprache markiert.

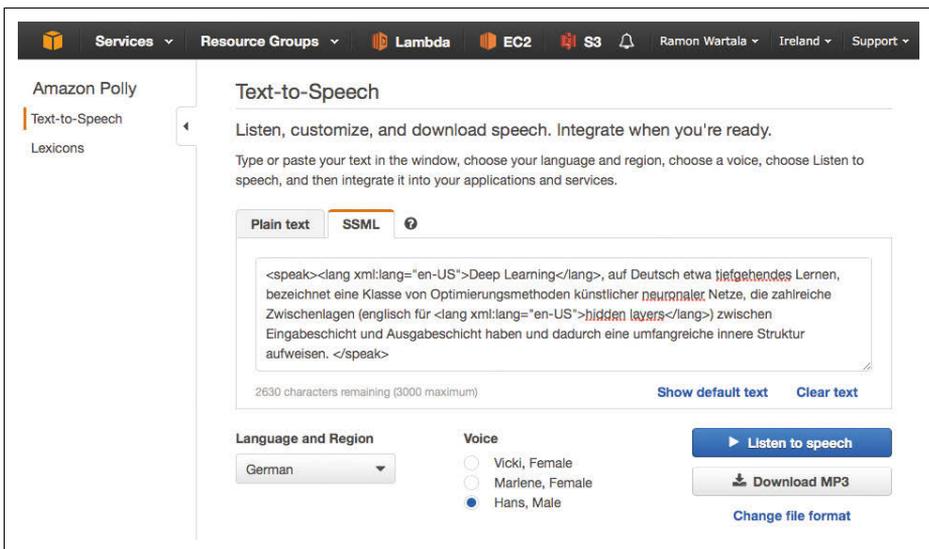


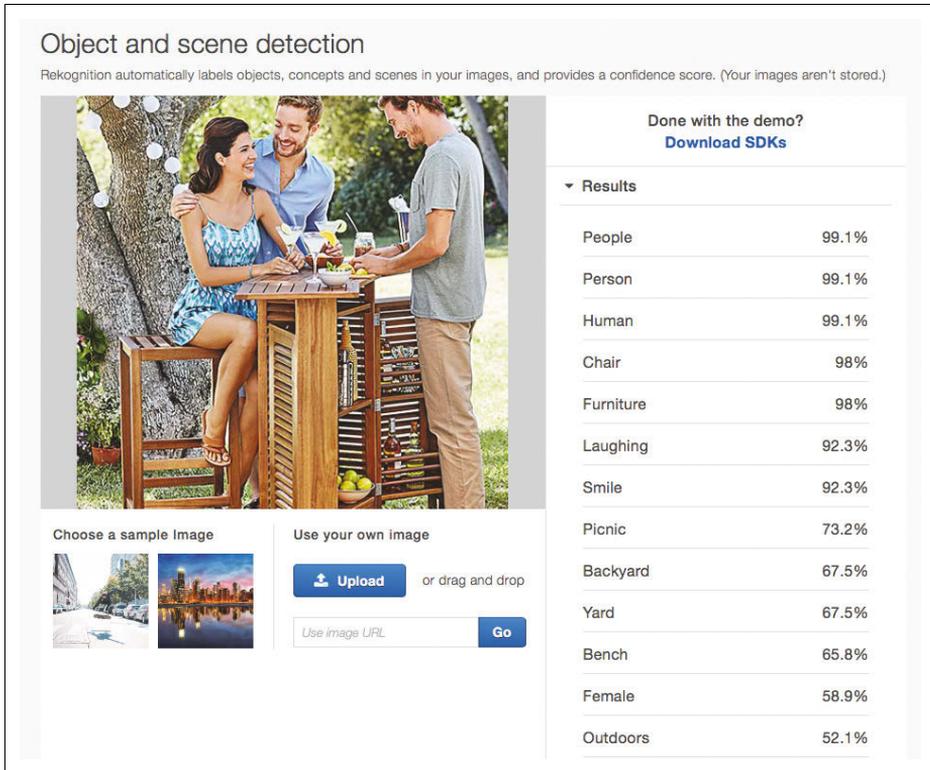
Abbildung 3-9: Amazon Polly innerhalb der AWS Management Console. Hier können einfache, aber auch mit SSML ausgezeichnete Texte eingegeben, angehört und als MP3, Ogg oder als PCM-Audiodatei heruntergeladen werden.

Um die Aussprache bestimmter Wörter zu verbessern, ermöglicht Polly die Definition eigener Aussprachelexika nach der Pronunciation Lexicon Specification PLS, die ein weiterer W3C-Standard ist (<https://www.w3.org/TR/pronunciation-lexicon/>). Als Ausgabeformat der Sprachsynthese kann dabei zwischen MP3, Ogg und PCM gewählt werden. Um Polly in eigene Anwendungen zu integrieren, kann Amazons

AWS-SDK für die Programmiersprachen Java, PHP, Python, Ruby, .NET oder Node.js genutzt werden.

Rekognition

Der AWS-Dienst *Rekognition* ermöglicht die Klassifikation von Bildern innerhalb der Amazon-Cloud. Auch hier ist die Nutzung über einen gültigen AWS-Account und das AWS-eigene Software Development Kit möglich. Die AWS Management Console bietet einen ersten Eindruck davon, wie sich der Dienst verhält. Abbildung 3-10 zeigt ein hochgeladenes Foto in der AWS Management Console und die von Rekognition erkannten Objekte.



The screenshot displays the 'Object and scene detection' interface. On the left, a photo of three people at a picnic is shown. Below it are options to 'Choose a sample image' or 'Use your own image' with an 'Upload' button and a 'Use image URL' field with a 'Go' button. On the right, a table lists detected objects and their confidence scores.

Object	Confidence Score
People	99.1%
Person	99.1%
Human	99.1%
Chair	98%
Furniture	98%
Laughing	92.3%
Smile	92.3%
Picnic	73.2%
Backyard	67.5%
Yard	67.5%
Bench	65.8%
Female	58.9%
Outdoors	52.1%

Abbildung 3-10: Amazon Rekognition ermöglicht die Erkennung von Objekten innerhalb eines Bilds.

Doch der Dienst kann noch mehr. So hilft Rekognition auch bei der Erkennung von Bildinhalten, die explizit oder suggestiv für Erwachsene gedacht sind. In Abbildung 3-11 ist die Bildklassifizierung eines Bikinis zu sehen. Rekognition erkennt darin ein suggestives Motiv mit einer Wahrscheinlichkeit von 76 %.

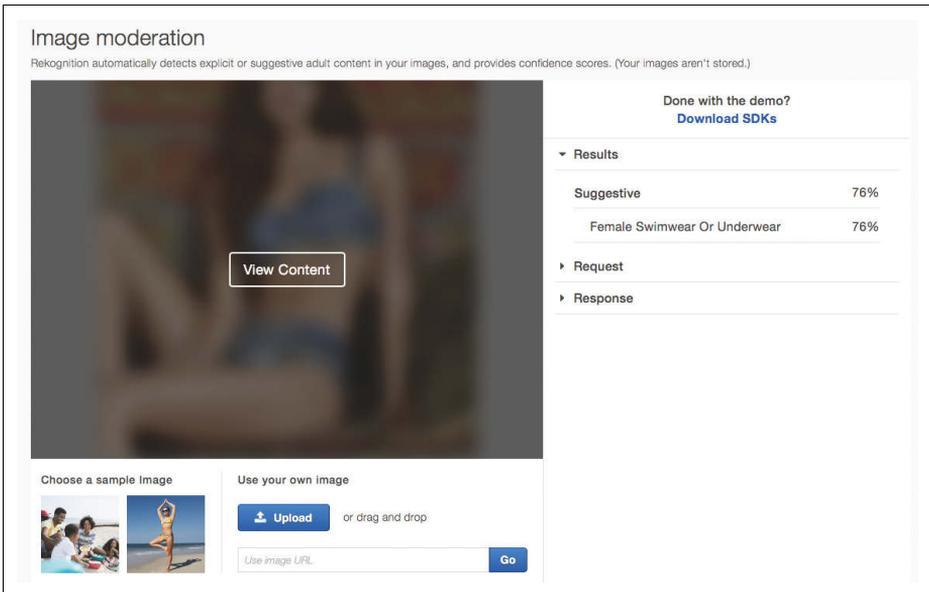


Abbildung 3-11: Amazon Rekognition kann auch dazu verwendet werden, Bilder mit expliziten oder suggestiven Elementen zu erkennen.

Der Rekognition-Dienst kann darüber hinaus ebenfalls erkennen, wo im Bild sich ein Gesicht befindet, und kann einen Rahmen darum zeichnen. Werden zwei Bilder mit Gesichtern dem Dienst übergeben, erkennt Rekognition, ob das Gesicht des einen Bilds im zweiten Bild vorhanden ist.

Lex

Amazon Lex ist einer der Dienste hinter Amazons Echo-Produkten. Lex ermöglicht die automatische Spracherkennung natürlicher Sprache. Ein mögliches Anwendungsfeld ist die Integration von Amazon Lex als Chatbot. Amazon selbst hat für diesen Zweck ein paar Demos eingerichtet, die die Erstellung von Chatbots zeigen (Abbildung 3-12). Zugänglich sind diese Demos über die AWS Management Console.

Diese Übersicht sollte nur einen ersten Einblick in die unterschiedlichen Produkte und Dienstleistungen geben, die bereits heute öffentlich nutzbar sind. Darüber hinaus gibt es zahlreiche weitere Firmen, die im Umfeld von Machine Learning und Deep Learning beratend tätig sind. Um eigene Deep-Learning-Anwendungen erstellen zu können, benötigt man neben der historischen Einordnung und dem Verständnis für mögliche Anwendungsfälle noch weiteres Know-how.

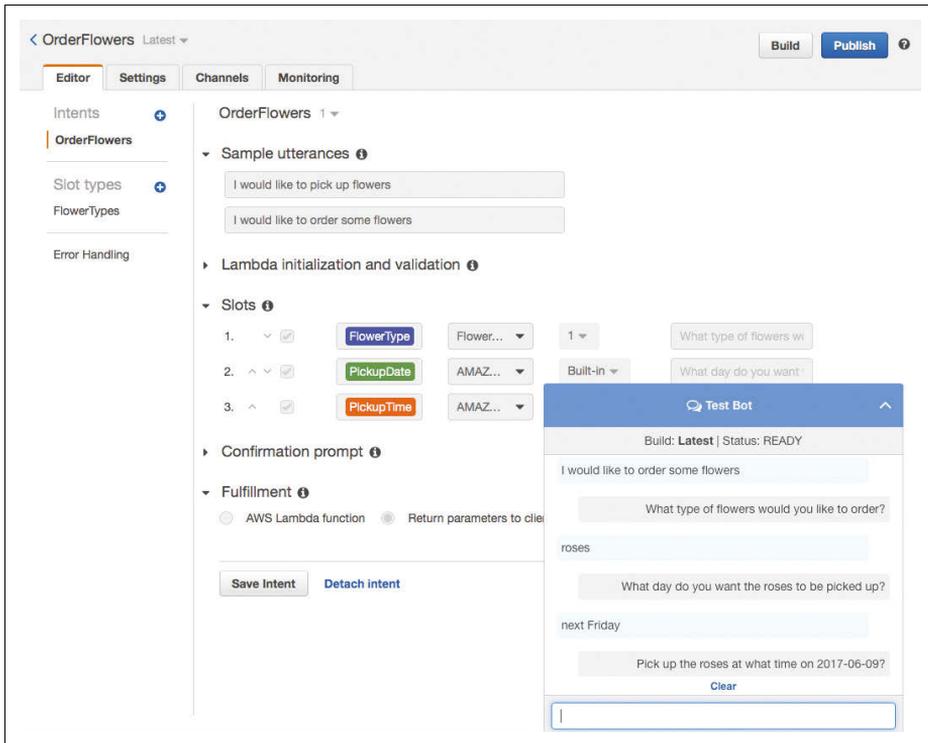


Abbildung 3-12: Amazon Lex enthält alles, um einen einfachen Chatbot aufzubauen. Das Beispiel implementiert einen Chatbot für die Bestellung von Blumen und ist Teil der Amazon-Produktdemonstration für Lex.

Im nächsten Kapitel wird eine »Werkbank« vorgestellt, die die unkomplizierte Nutzung der Deep-Learning-Frameworks nicht nur innerhalb dieses Buchs ermöglicht. Wie sich diese unter Windows, Linux und macOS installieren lässt, erfahren Sie jetzt.