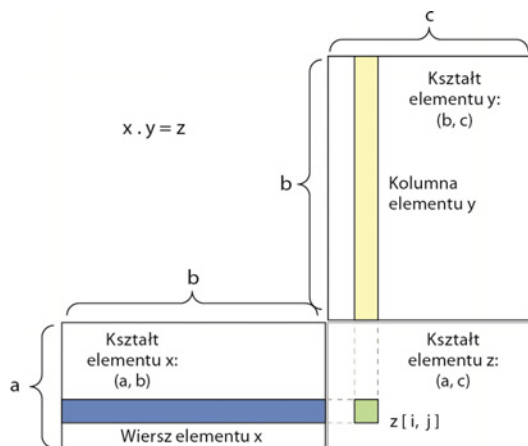


# Deep Learning.

## Praca z językiem Python i biblioteką Keras

---

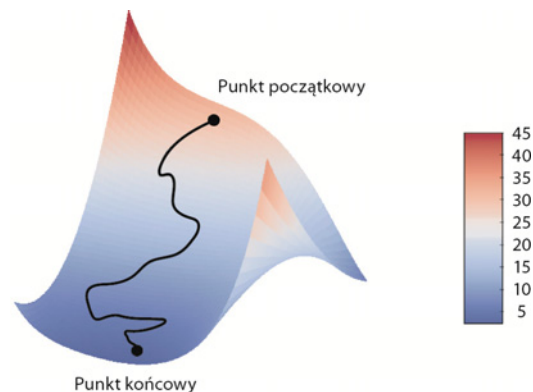
### Rozdział 2.



**Rysunek 2.5.** Schemat obliczania iloczynu skalarnego macierzy

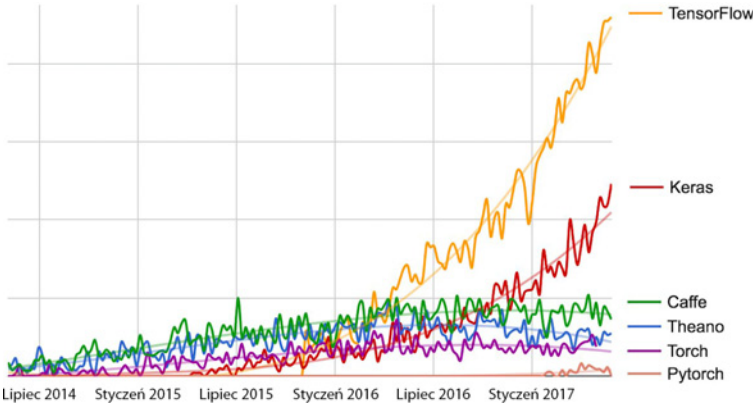


**Rysunek 2.9.** Rozkładanie skomplikowanego zbioru danych

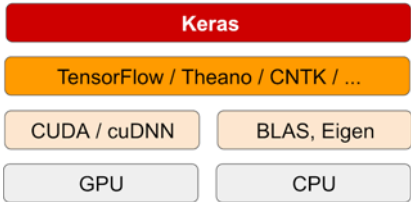


**Rysunek 2.12.** Spadek gradientowy w dwuwymiarowej przestrzeni straty (sieć trenuje dwa parametry)

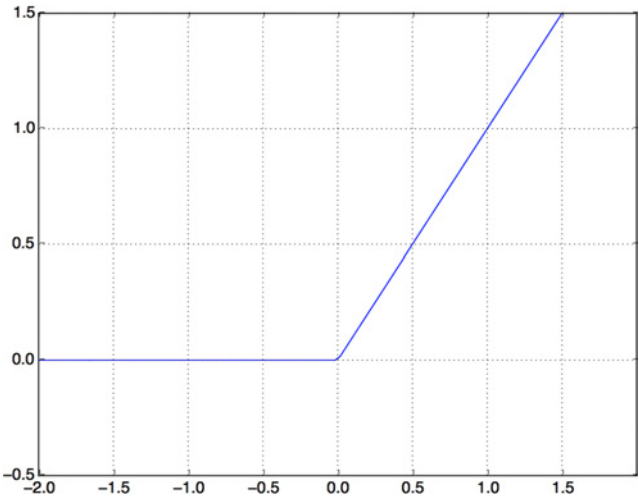
### Rozdział 3.



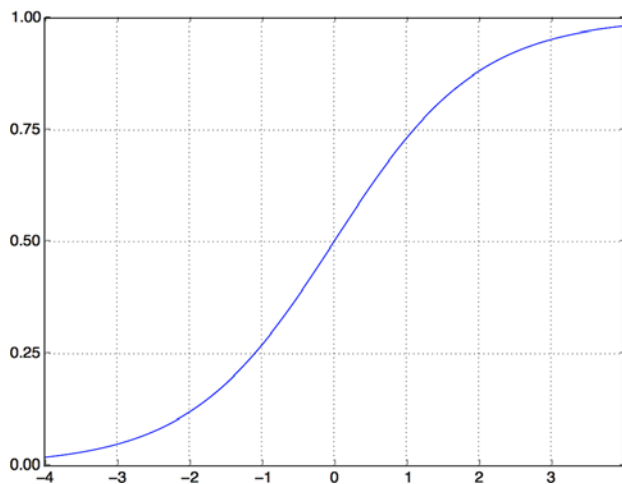
**Rysunek 3.2.** Wykres zmian zainteresowania różnymi pakietami uczenia maszynowego sporządzony na podstawie danych wyszukiwarki Google



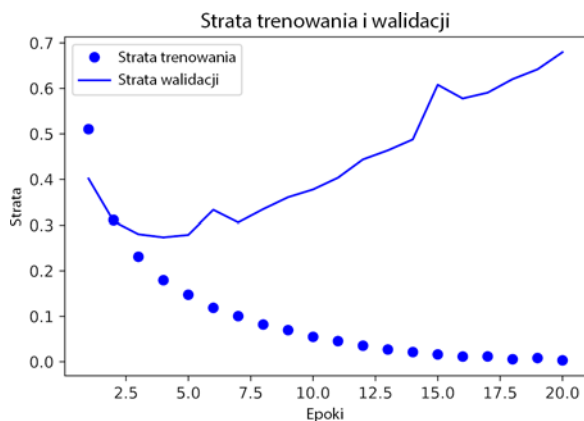
**Rysunek 3.3.** Stos programowy i sprzętowy uczenia głębokiego



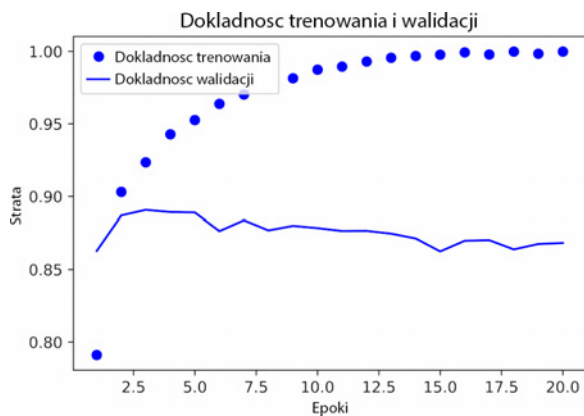
**Rysunek 3.4.** Funkcja relu



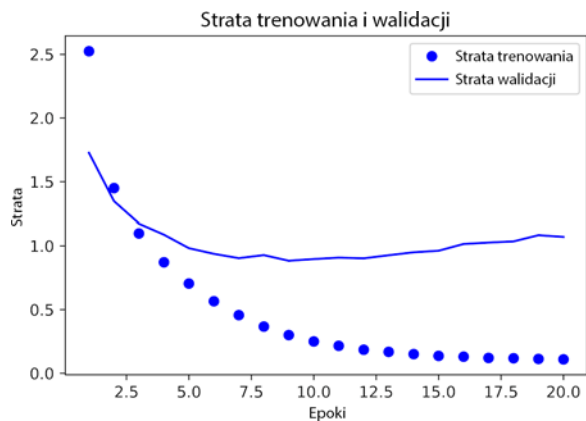
Rysunek 3.5. Funkcja sigmoid



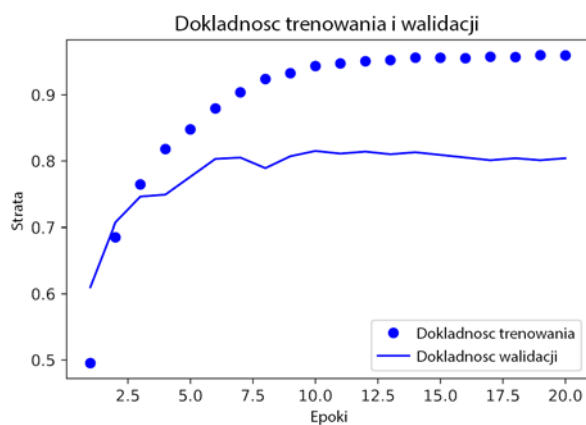
Rysunek 3.7. Strata trenowania i walidacji



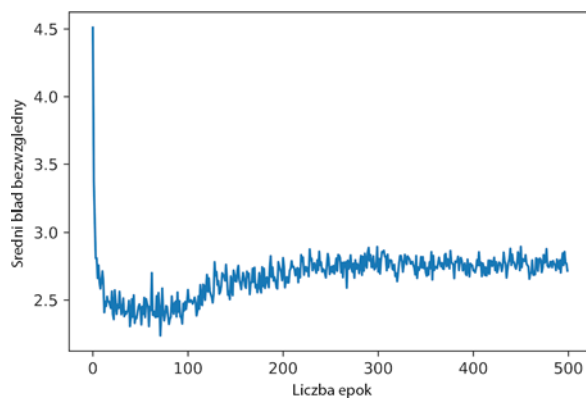
Rysunek 3.8. Dokładność trenowania i walidacji



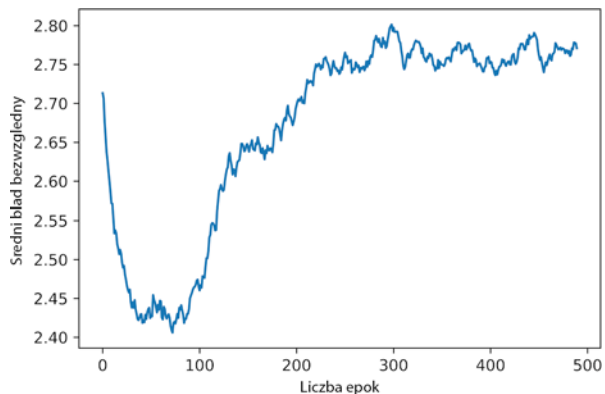
**Rysunek 3.9.** Strata trenowania i walidacji



**Rysunek 3.10.** Dokładność trenowania i walidacji

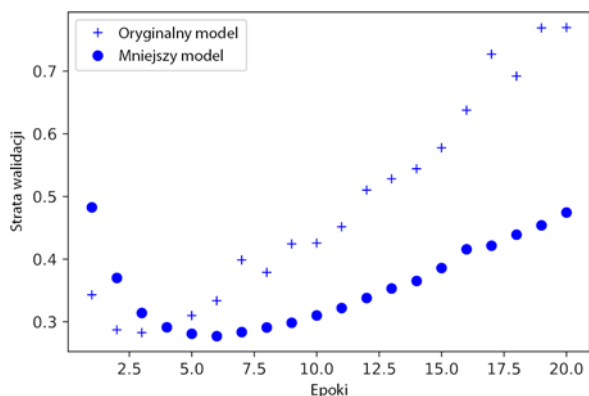


**Rysunek 3.12.** Średni błąd bezwzględny w poszczególnych epokach

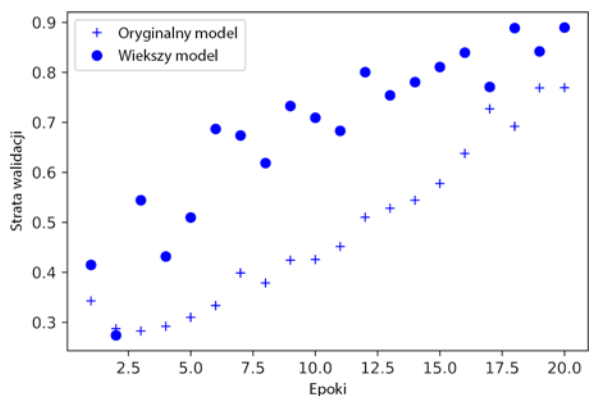


**Rysunek 3.13.** Średni błąd bezwzględny w poszczególnych epokach (bez 10 pierwszych obserwacji)

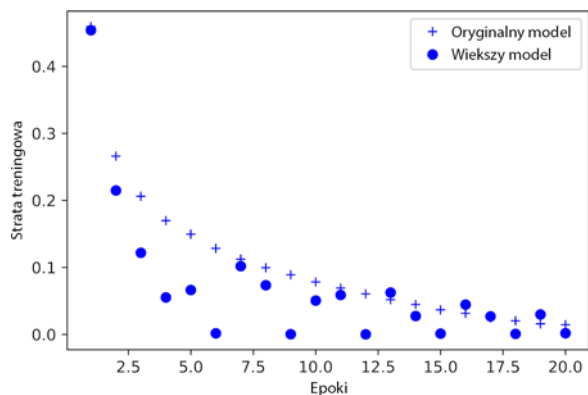
## Rozdział 4.



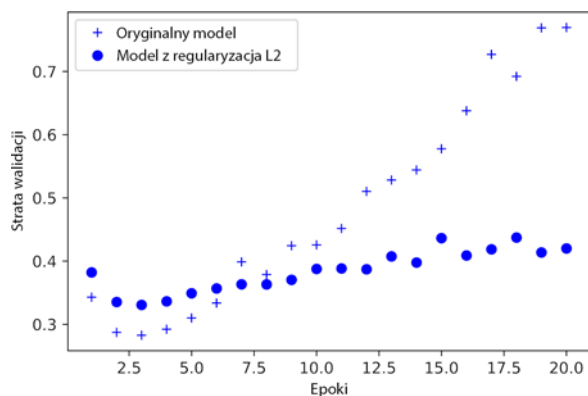
**Rysunek 4.4.** Wpływ pojemności modelu na stratę walidacji: działanie mniejszego modelu



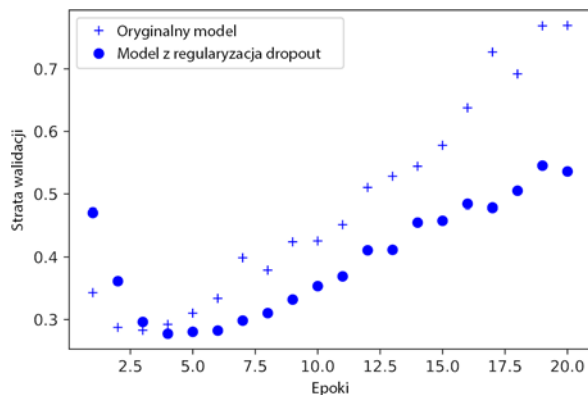
**Rysunek 4.5.** Wpływ pojemności modelu na stratę walidacji: działanie większego modelu



**Rysunek 4.6.** Wpływ pojemności modelu na stratę treningową: działanie większego modelu



**Rysunek 4.7.** Wpływ regularyzacji L2 na stratę walidacji

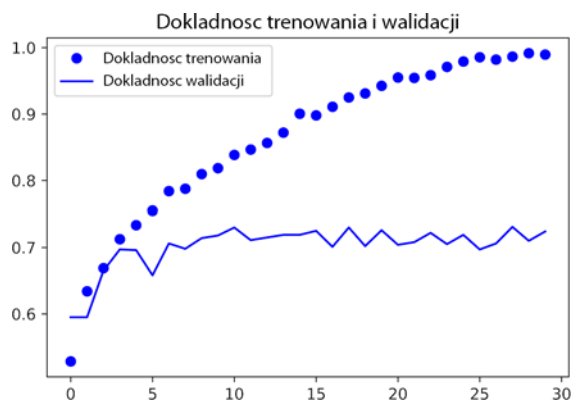


**Rysunek 4.9.** Wpływ zastosowania regularyzacji metodą porzucania na zmianę wartości straty walidacji

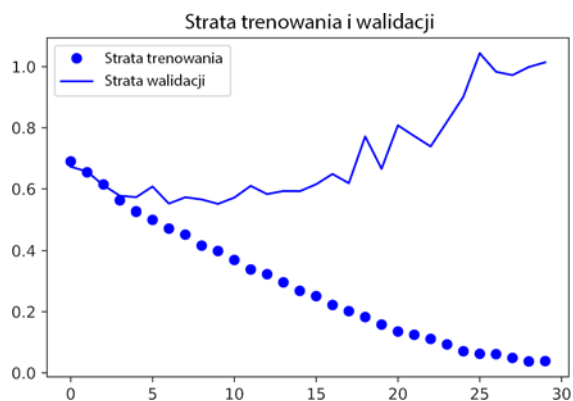
## Rozdział 5.



**Rysunek 5.8.** Przykładowe zdjęcia wchodzące w skład zbioru „Dogs vs. Cats”; nie zmieniałem ich rozmiarów: próbki są heterogeniczne, jeżeli chodzi o rozmiar, wygląd itd.



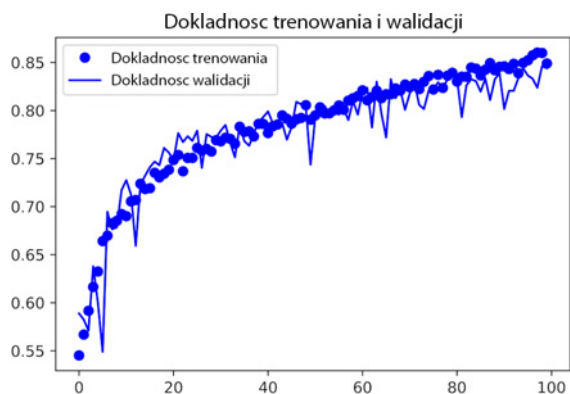
**Rysunek 5.9.** Dokładność trenowania i walidacji



**Rysunek 5.10.** Strata trenowania i walidacji



**Rysunek 5.11.**  
Generowanie zdjęć  
kotów poprzez losowe  
modyfikowanie obrazu

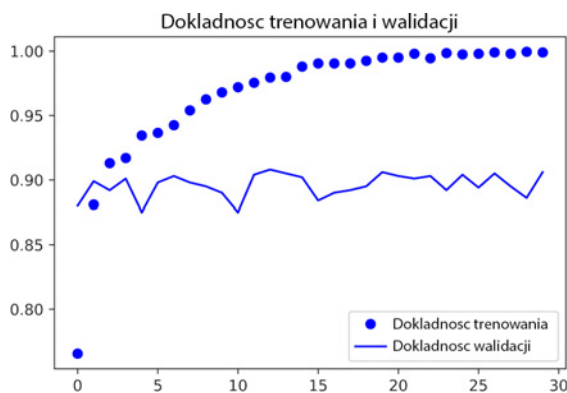


**Rysunek 5.12.** Dokładność trenowania  
i walidacji po zastosowaniu techniki  
augmentacji danych

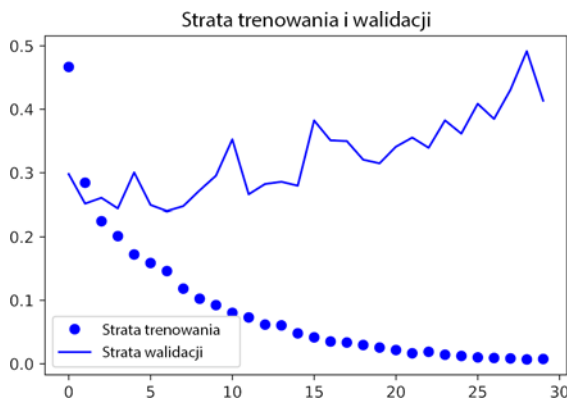




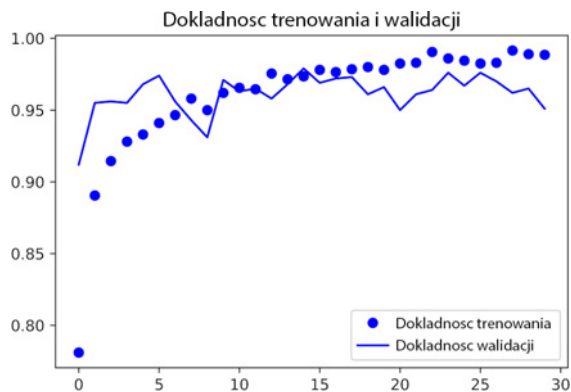
**Rysunek 5.13.** Strata trenowania i walidacji po zastosowaniu techniki augmentacji danych



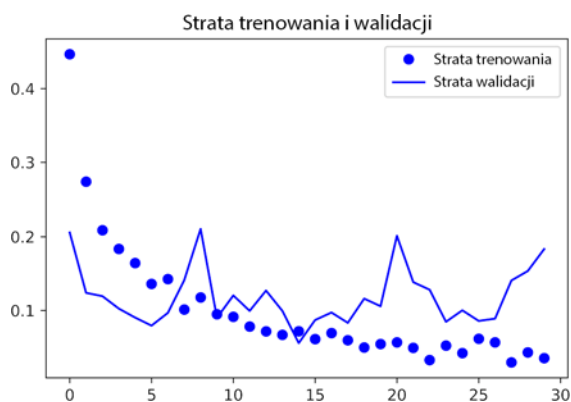
**Rysunek 5.15.** Dokładność trenowania i walidacji po zastosowaniu techniki prostej ekstrakcji cech



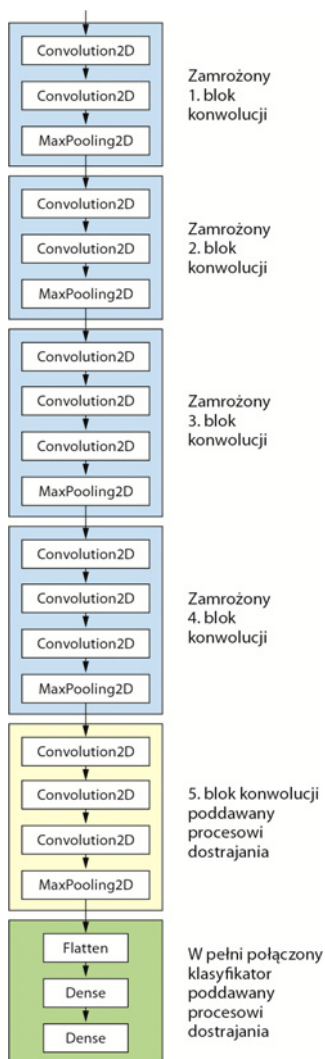
**Rysunek 5.16.** Strata trenowania i walidacji po zastosowaniu techniki prostej ekstrakcji cech



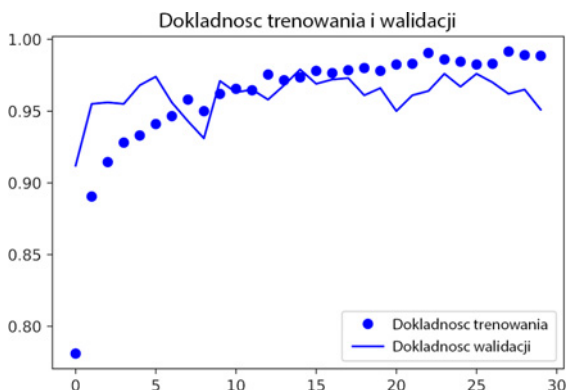
**Rysunek 5.17.** Dokładność trenowania i walidacji ekstrakcji cech przy zastosowaniu techniki augmentacji danych



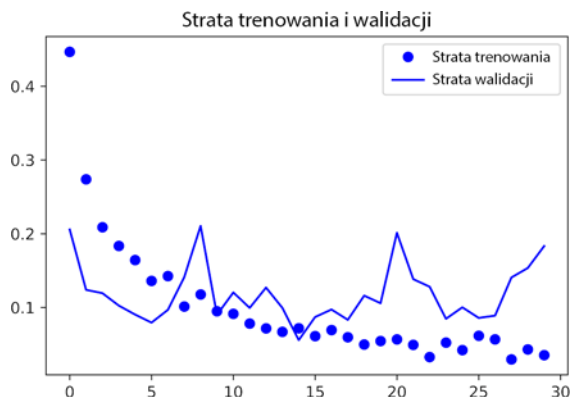
**Rysunek 5.18.** Strata trenowania i walidacji ekstrakcji cech przy zastosowaniu techniki augmentacji danych



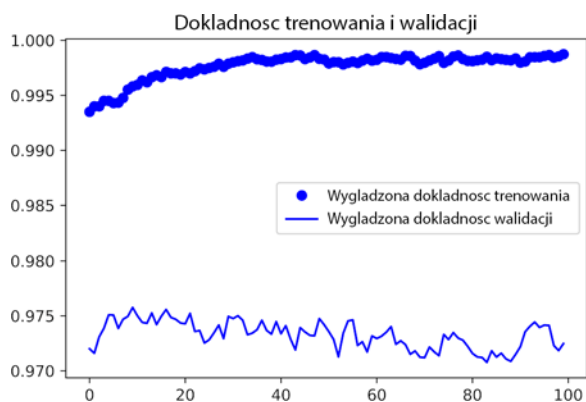
**Rysunek 5.19.** Dostrajanie ostatniego bloku konwolucji sieci VGG16



**Rysunek 5.20.** Dokładność trenowania i walidacji podczas dostrajania



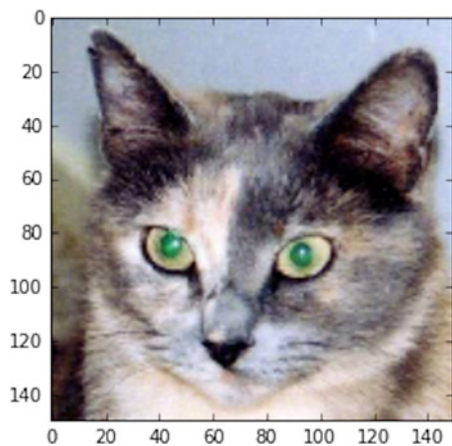
**Rysunek 5.21.** Strata trenowania i walidacji podczas dostrajania



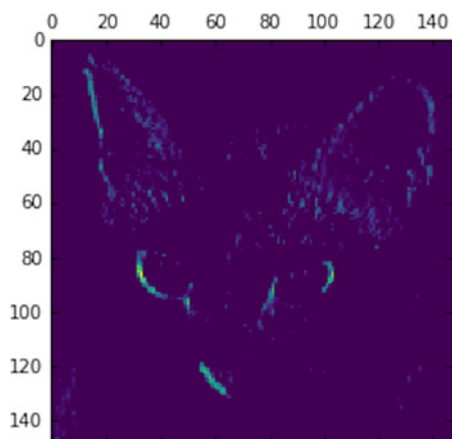
**Rysunek 5.22.** Wyglądzone krzywe dokładności trenowania i walidacji podczas dostrajania



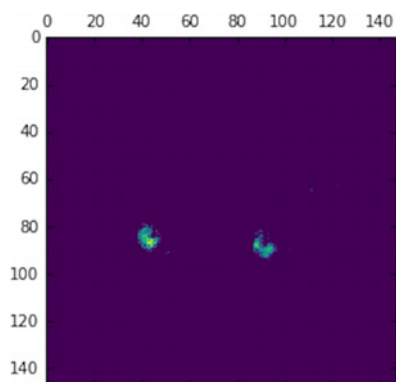
**Rysunek 5.23.** Wyglądzone krzywe straty trenowania i walidacji podczas dostrajania



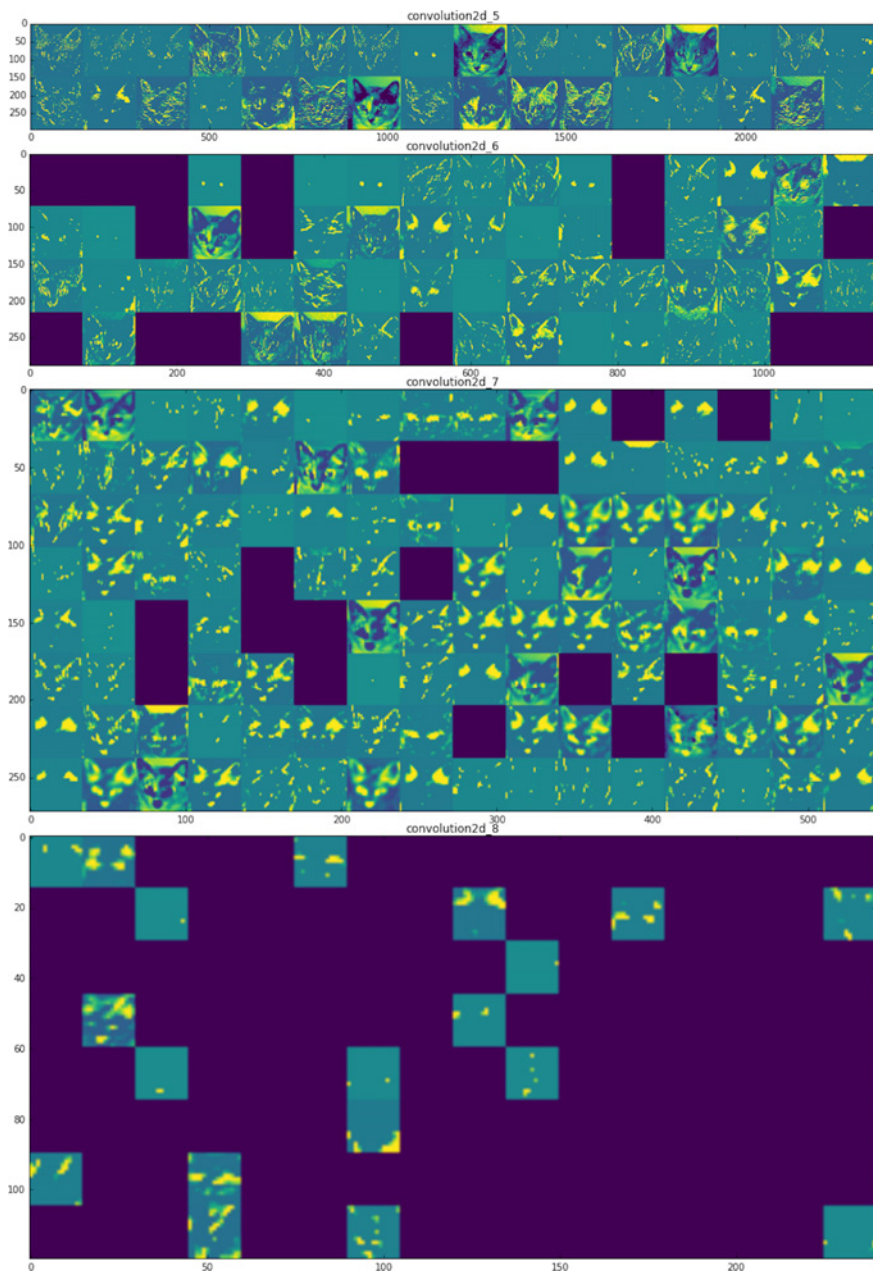
**Rysunek 5.24.** Testowy obraz kota



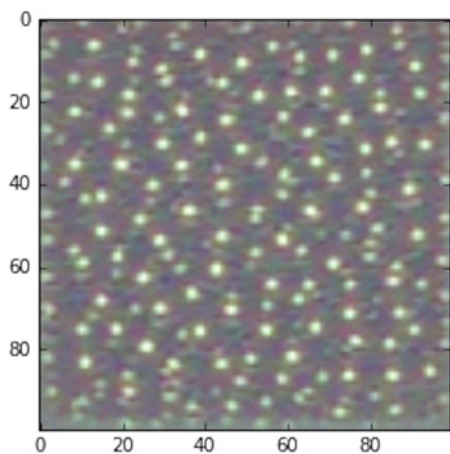
**Rysunek 5.25.** Czwarty kanał aktywacji pierwszej warstwy podczas przetwarzania testowego zdjęcia kota



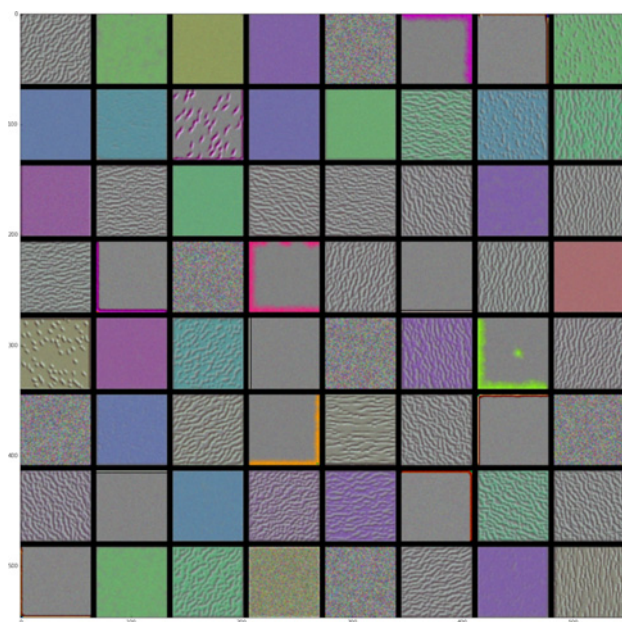
**Rysunek 5.26.** Siódmy kanał aktywacji pierwszej warstwy podczas przetwarzania testowego zdjęcia kota



**Rysunek 5.27.** Wszystkie kanały wszystkich warstw aktywacji podczas przetwarzania testowego zdjęcia kota

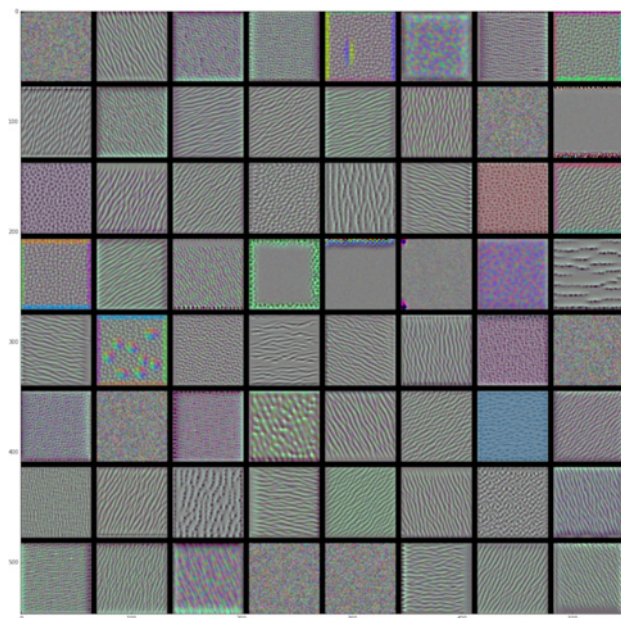


**Rysunek 5.29.** Wzór maksymalizujący odpowiedź zerowego kanału warstwy block3\_conv1

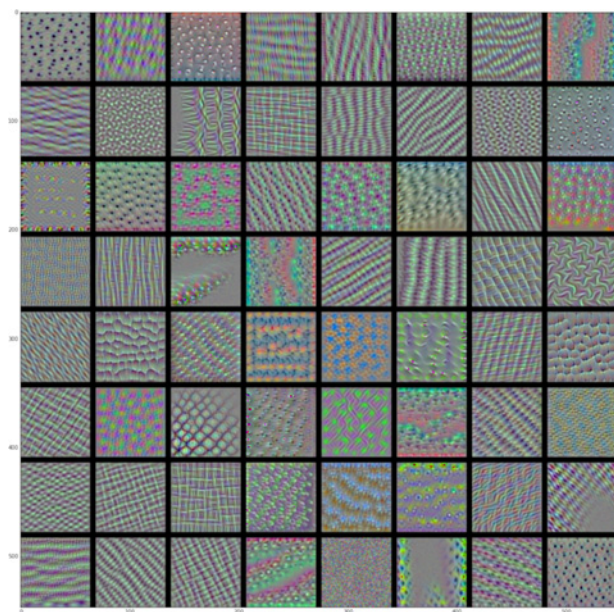


**Rysunek 5.30.** Wzorce filtrów warstwy block1\_conv1



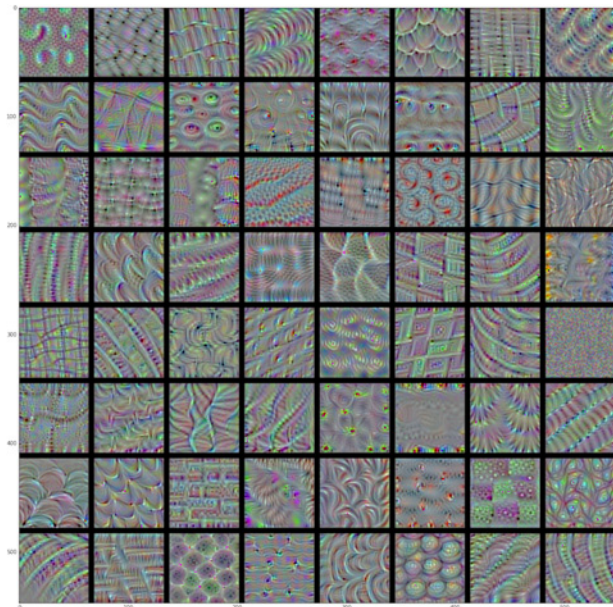


**Rysunek 5.31.** Wzorce filtrów warstwy block2\_conv1



**Rysunek 5.32.** Wzorce filtrów warstwy block3\_conv1

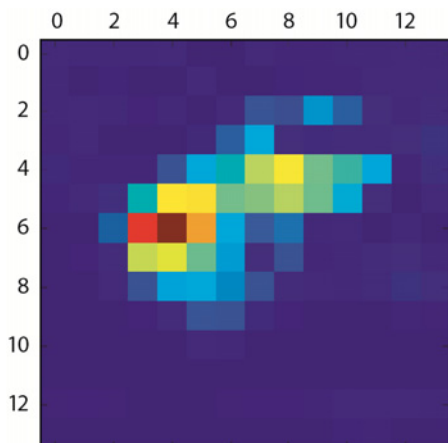




**Rysunek 5.33.** Wzorce filtrów warstwy block4\_conv1



**Rysunek 5.34.** Testowe zdjęcie afrykańskich słońi



**Rysunek 5.35.** Mapa ciepła aktywacji klasy słonia afrykańskiego przez obraz wejściowy

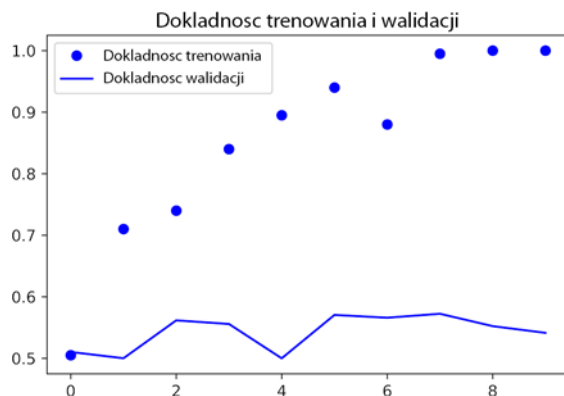


**Rysunek 5.36.** Nakładanie mapy ciepła na oryginalny obraz

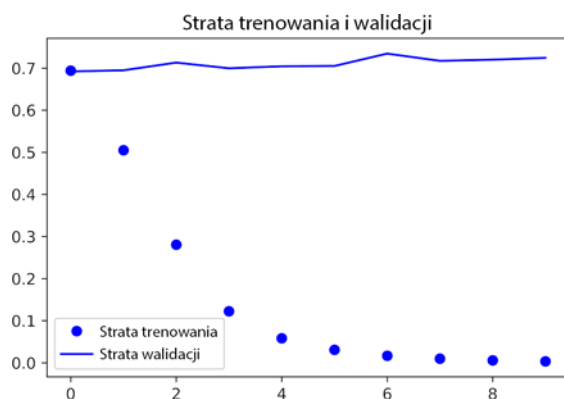
## Rozdział 6.



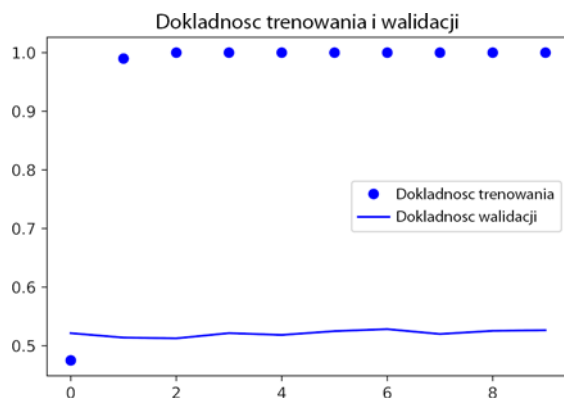
**Rysunek 6.5.** Strata trenowania i walidacji podczas korzystania z wytrenowanych wcześniej osadzeń słów



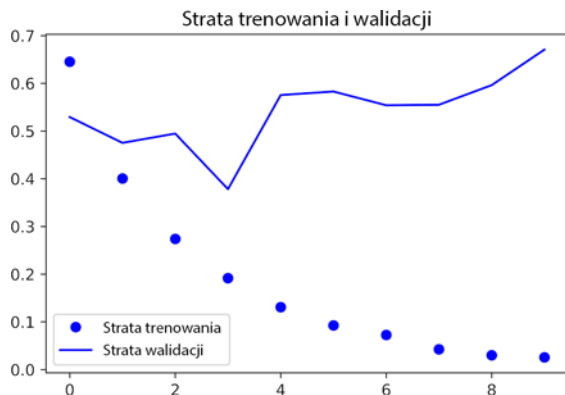
**Rysunek 6.6.** Dokładność trenowania i walidacji podczas korzystania z wytrenowanych wcześniej osadzeń słów



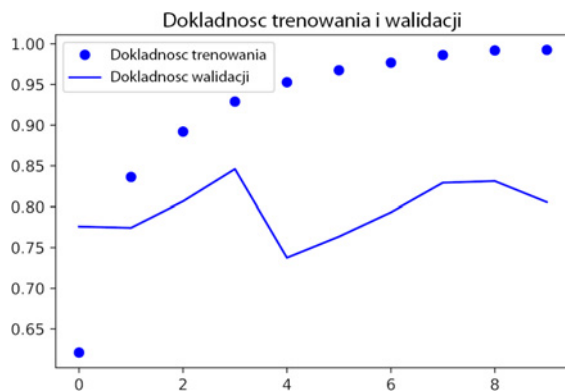
**Rysunek 6.7.** Strata trenowania i walidacji bez korzystania z wytrenowanych wcześniej osadzeń słów



**Rysunek 6.8.** Dokładność trenowania i walidacji bez korzystania z wytrenowanych wcześniej osadzeń słów



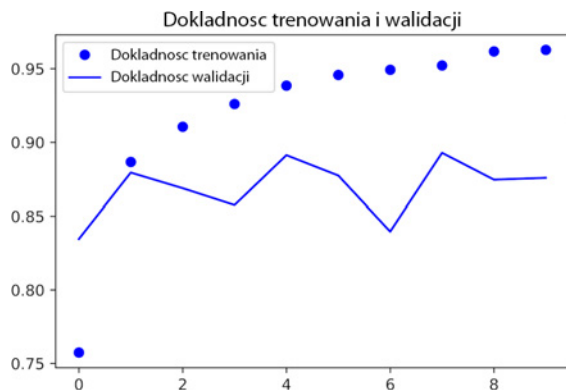
**Rysunek 6.11.** Strata trenowania i walidacji podczas przetwarzania zbioru IMDB przy użyciu warstwy SimpleRNN



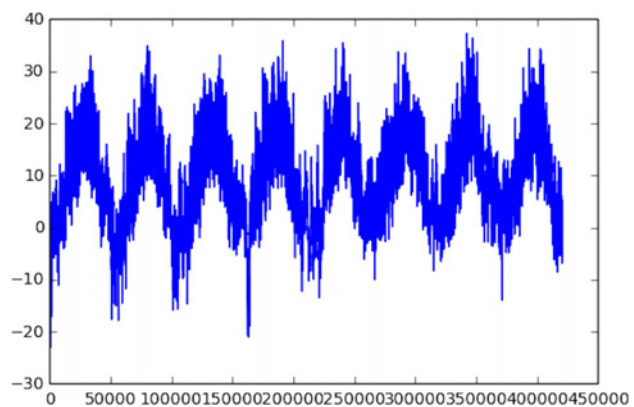
**Rysunek 6.12.** Dokładność trenowania i walidacji podczas przetwarzania zbioru IMDB przy użyciu warstwy SimpleRNN



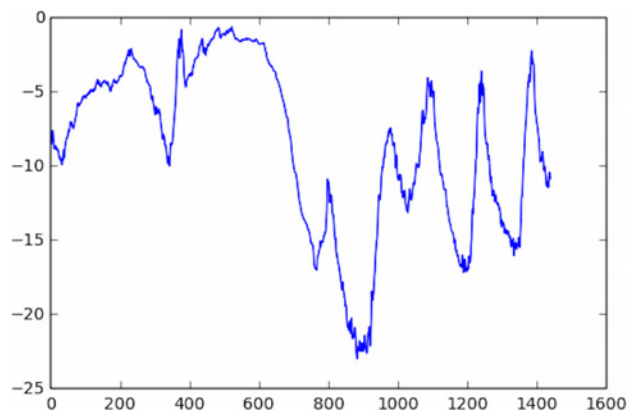
**Rysunek 6.16.** Strata trenowania i walidacji podczas przetwarzania zbioru IMDB przy użyciu warstwy LSTM



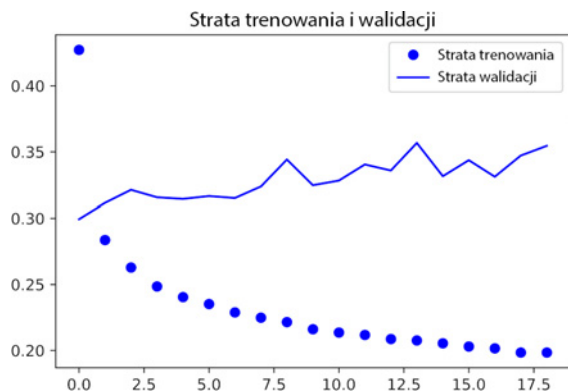
**Rysunek 6.17.** Dokładność trenowania i walidacji podczas przetwarzania zbioru IMDB przy użyciu warstwy LSTM



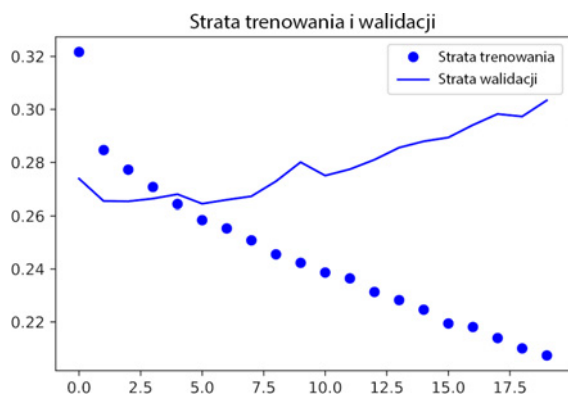
**Rysunek 6.18.** Zmiany temperatury w zakresie całego zbioru danych (°C)



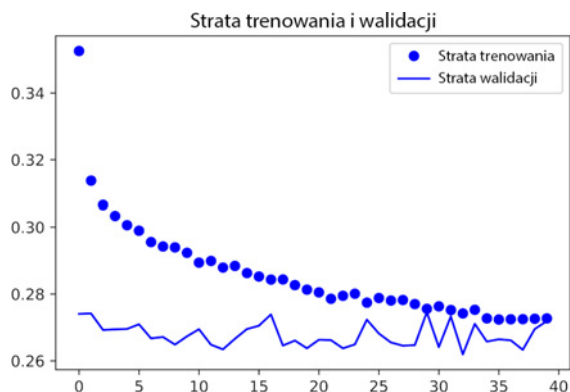
**Rysunek 6.19.** Zmiany temperatury w ciągu 10 pierwszych dni (°C)



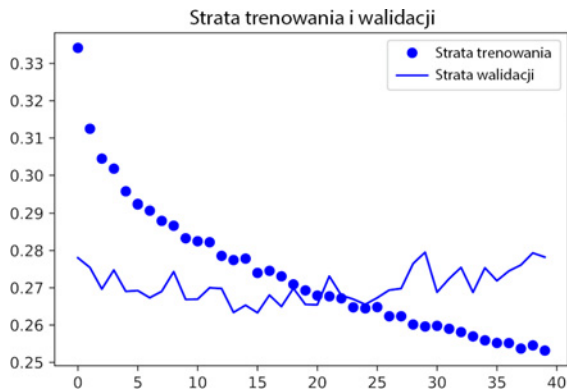
**Rysunek 6.20.** Strata trenowania i walidacji podczas przetwarzania zbioru danych temperatur Jena w ramach próby prognozowania za pomocą prostej, gęsto połączonej sieci



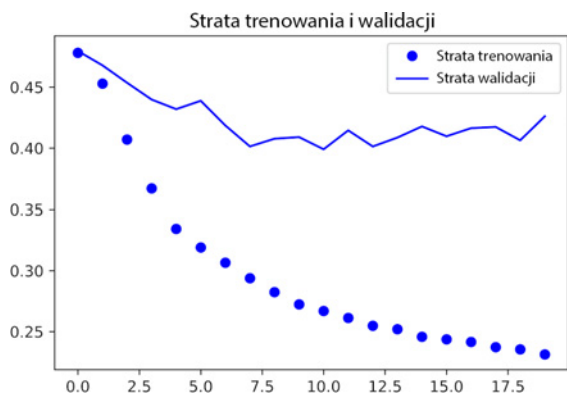
**Rysunek 6.21.** Strata trenowania i walidacji podczas przetwarzania zbioru danych temperatur Jena w ramach próby prognozowania za pomocą warstwy GRU



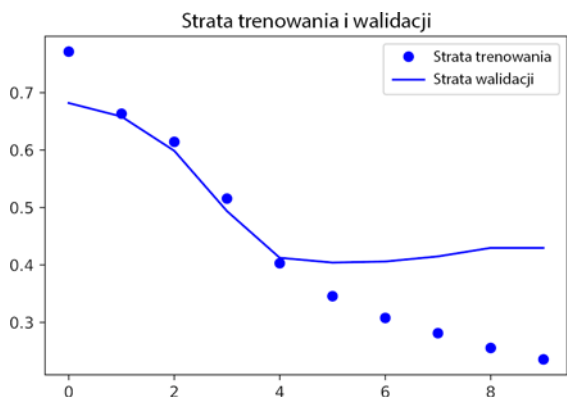
**Rysunek 6.22.** Strata trenowania i walidacji podczas przetwarzania zbioru danych temperatur Jena w ramach próby prognozowania za pomocą warstwy GRU przy regularyzacji poprzez odrzucanie



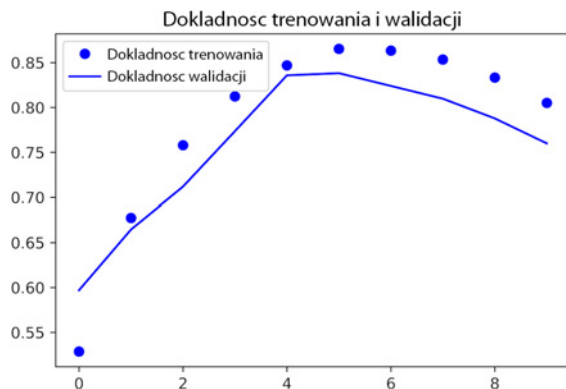
**Rysunek 6.23.** Strata trenowania i walidacji podczas przetwarzania zbioru danych temperatur Jena w ramach próby prognozowania za pomocą stosu warstw GRU



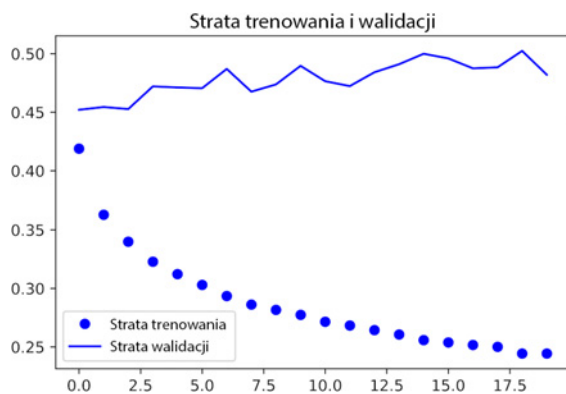
**Rysunek 6.24.** Strata trenowania i walidacji podczas przetwarzania zbioru danych temperatur Jena w ramach próby prognozowania za pomocą warstwy GRU przy odwróceniu kolejności sekwencji



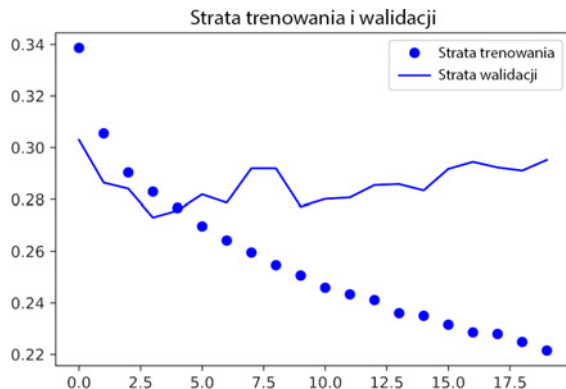
**Rysunek 6.27.** Strata trenowania i walidacji podczas przetwarzania zbioru danych IMDB przez prostą jednowymiarową sieć konwulcyjną



**Rysunek 6.28.** Dokładność trenowania i walidacji podczas zbioru danych IMDB przez prostą jednowymiarową sieć konwolucyjną



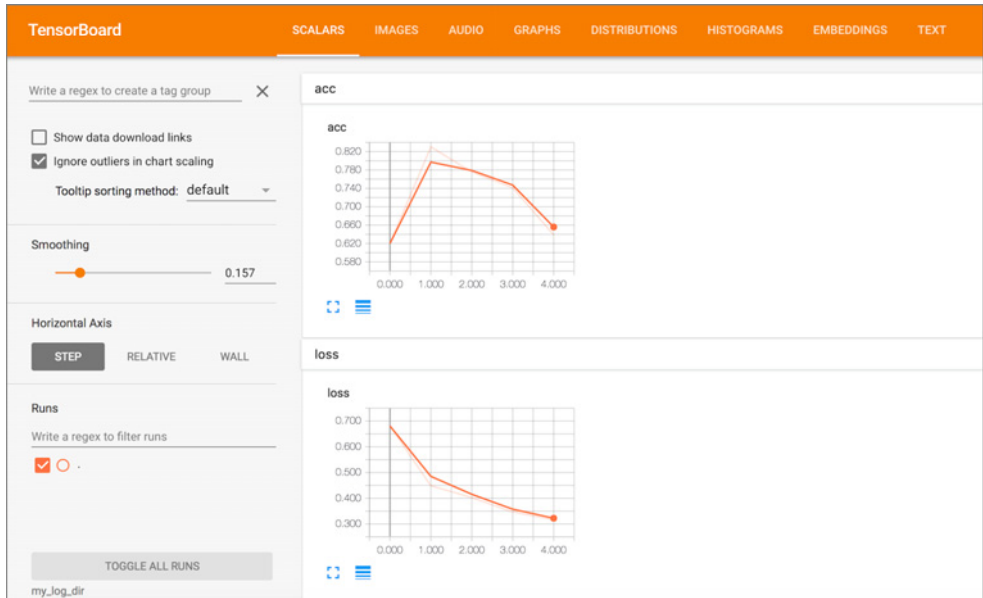
**Rysunek 6.29.** Strata trenowania i walidacji podczas przetwarzania zbioru danych Jena przez prostą jednowymiarową sieć konwolucyjną



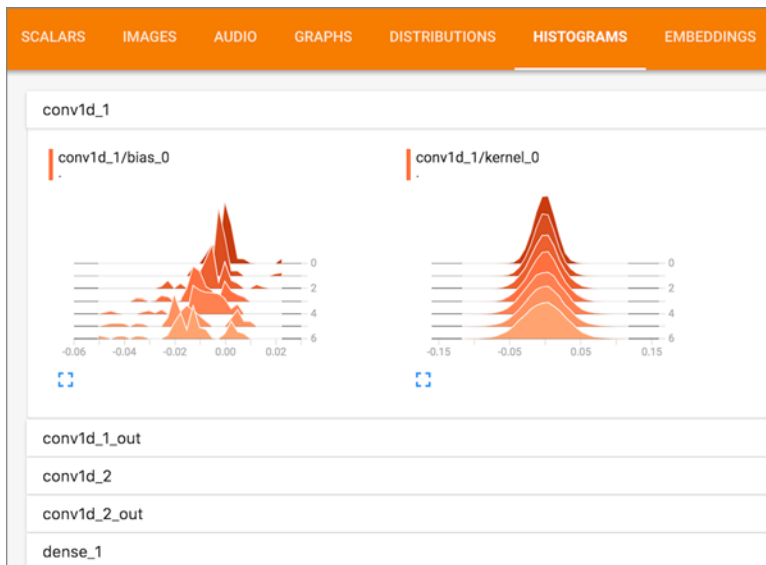
**Rysunek 6.31.** Strata trenowania i walidacji podczas przetwarzania zbioru danych Jena przez prostą jednowymiarową sieć konwolucyjną połączoną z warstwą GRU



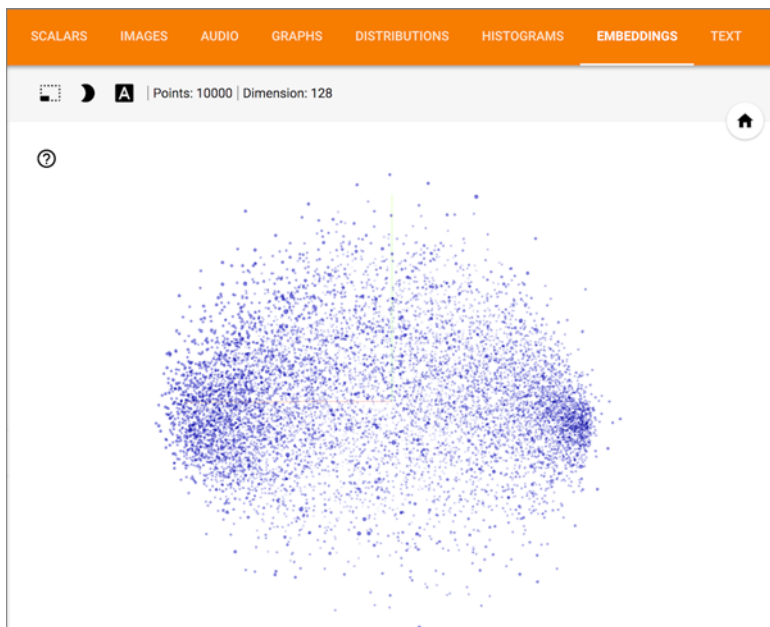
## Rozdział 7.



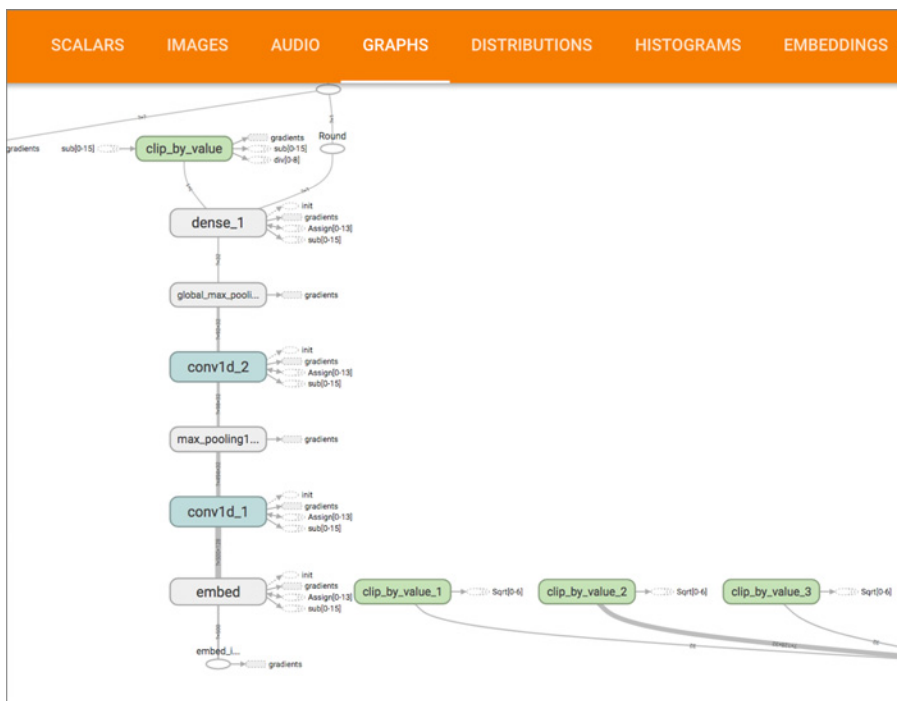
Rysunek 7.10. Monitorowanie metryk modelu za pomocą narzędzia TensorBoard



Rysunek 7.11. Histogramy aktywacji wyświetlane przez narzędzie TensorBoard

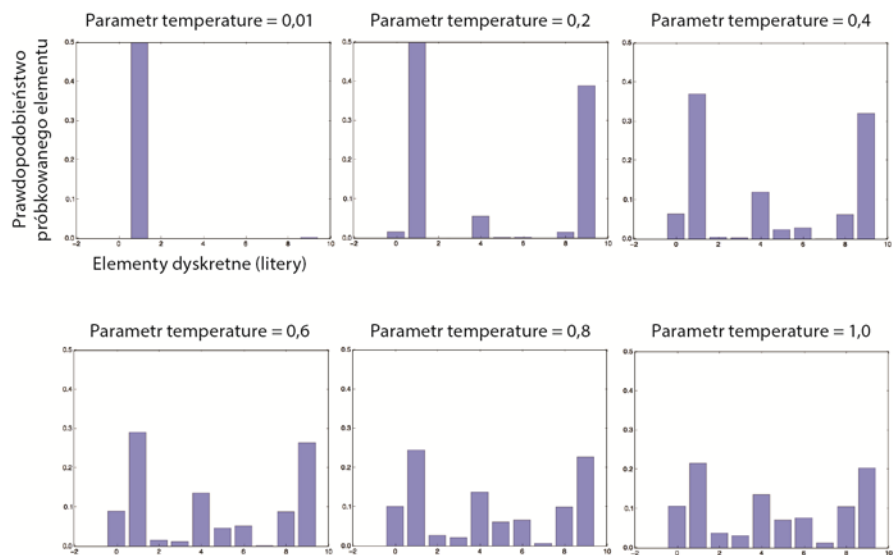


**Rysunek 7.12.** Interaktywna trójwymiarowa wizualizacja osadzeń słów wygenerowana przez narzędzie TensorBoard

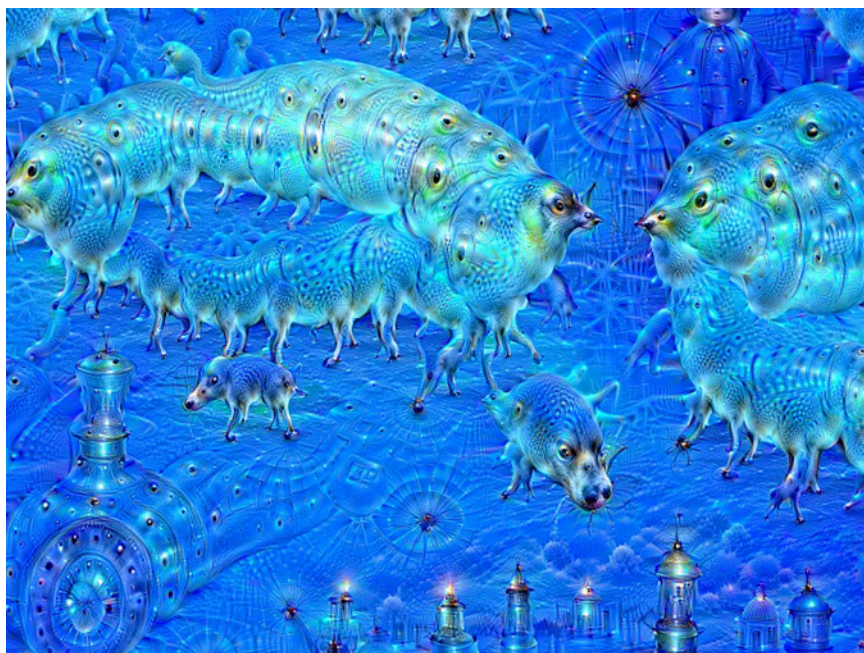


**Rysunek 7.13.** Wizualizacja działania mechanizmów biblioteki TensorFlow wygenerowana przy użyciu narzędzia TensorBoard

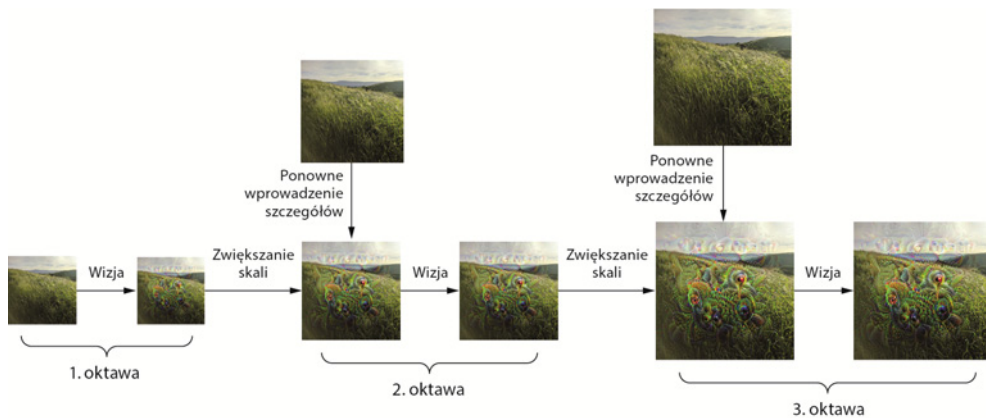
## Rozdział 8.



**Rysunek 8.2.** Ten sam rozkład prawdopodobieństwa z różnymi wagami; zmniejszenie parametru temperature powoduje większą przewidywalność; zwiększenie parametru temperature zwiększa losowość



**Rysunek 8.3.** Przykładowy obraz wygenerowany przez algorytm DeepDream



**Rysunek 8.4.** Działanie algorytmu DeepDream: następujące po sobie operacje skalowania przestrzennego (oktawy) i dodawania szczegółów



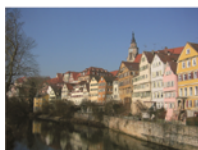
**Rysunek 8.5.** Przetwarzanie przykładowego obrazu za pomocą algorytmu DeepDream





**Rysunek 8.6.** Przetwarzanie przykładowego obrazu przez różne konfiguracje algorytmu DeepDream

Obraz poddawany  
modyfikacji



+

Referencyjny styl



=

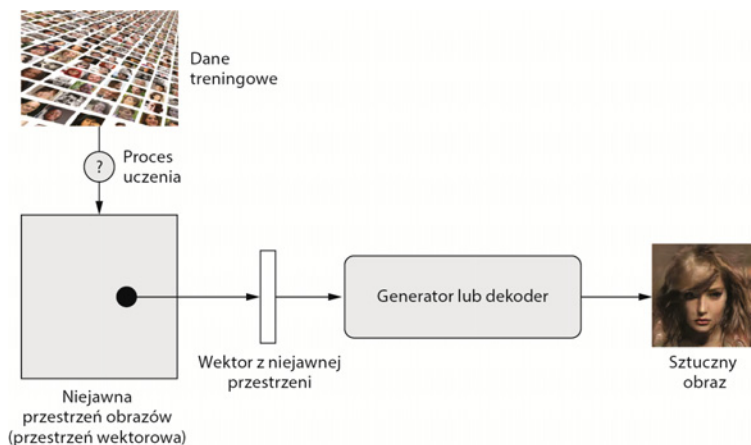
Zmodyfikowany obraz



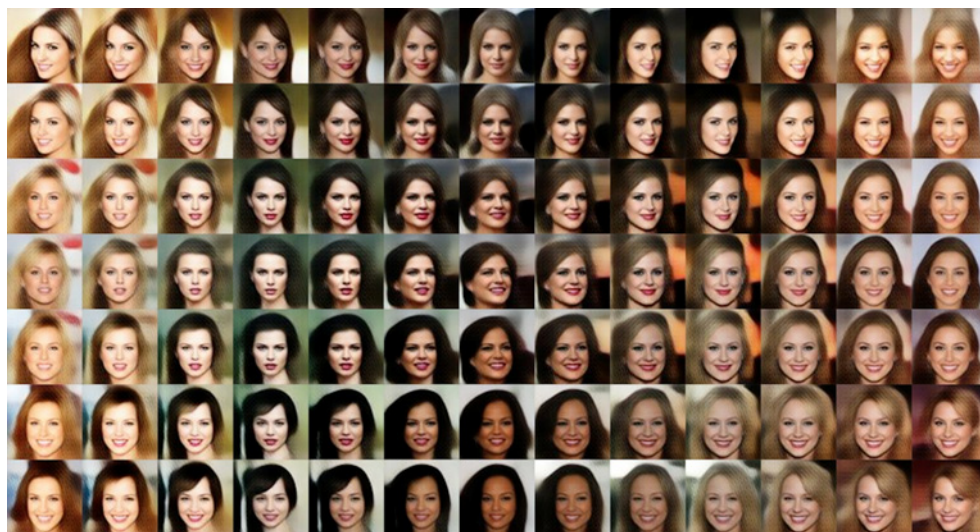
**Rysunek 8.7.** Przykład transferu stylu



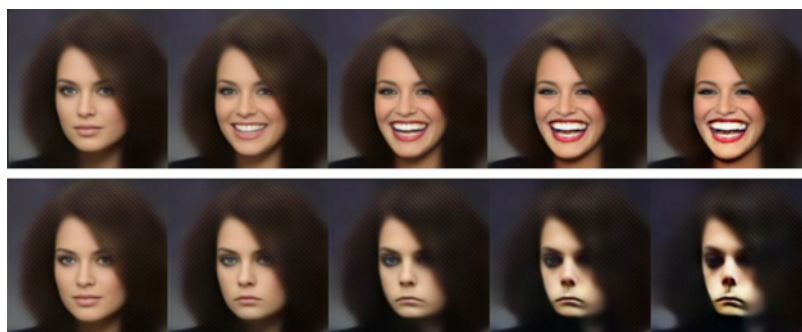
**Rysunek 8.8.** Przykładowe efekty pracy algorytmu



**Rysunek 8.9.** Trenowanie niejawnej wektorowej przestrzeni obrazów i używanie jej do próbkowania nowych obrazów

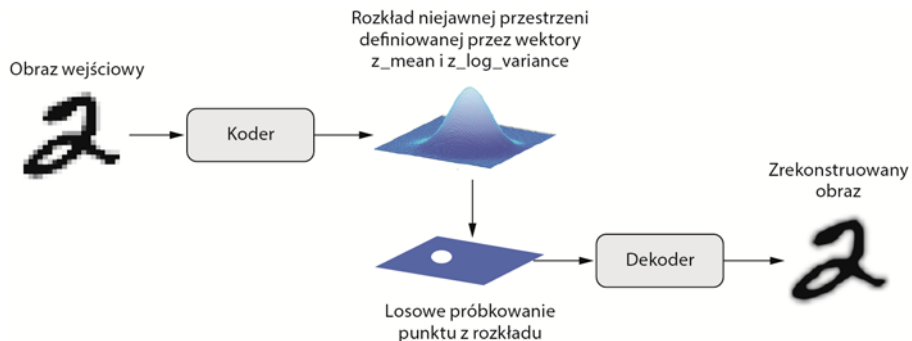


**Rysunek 8.10.** Ciągła przestrzeń twarzy wygenerowana przez Toma White'a przy użyciu koderów VAE

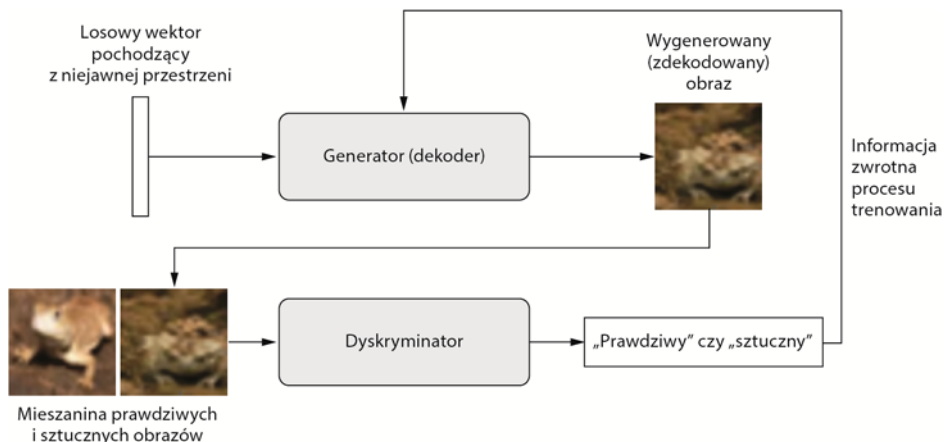


**Rysunek 8.11.** Wektor uśmiechu





**Rysunek 8.13.** Koder VAE mapuje obraz na dwa wektory:  $z\_mean$  i  $z\_log\_variance$ , które definiują rozkład prawdopodobieństwa w niejawnej przestrzeni używanej do próbkowania dekodowanego punktu

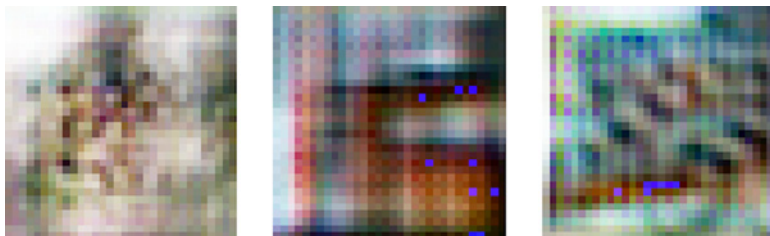


**Rysunek 8.15.** Generator zamienia losowe wektory niejawnej przestrzeni w obrazy, a dyskryminator stara się odróżnić prawdziwe obrazy od tych, które zostały wygenerowane; generator jest trenowany w celu oszukania dyskryminatora

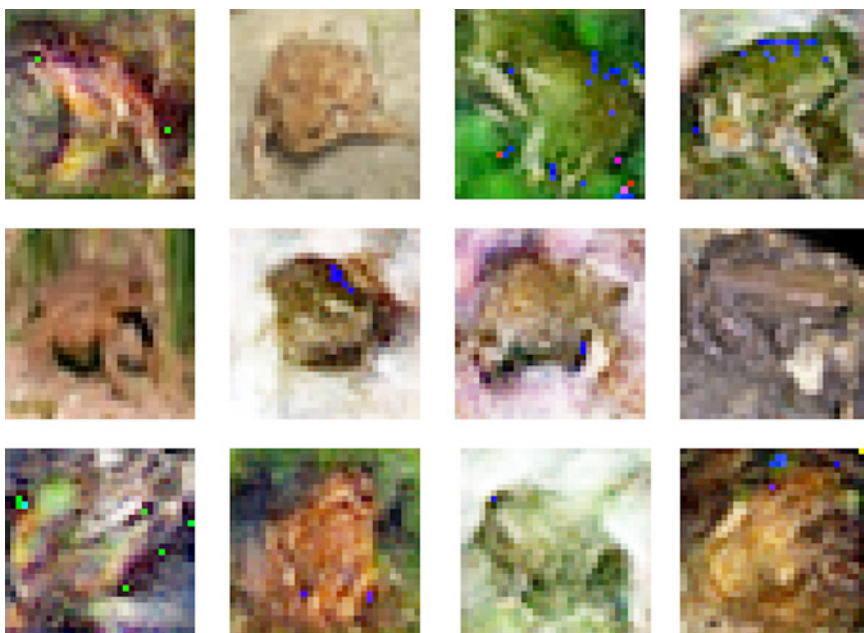


**Rysunek 8.16.** Obrazy wyciągnięte z niejawnej przestrzeni przez sieć GAN; zostały wygenerowane przez Mike'a Tykę przy użyciu wieloetapowej sieci GAN trenowanej na zbiorze danych ze zdjęciami twarzy (<http://www.miketyka.com/>)



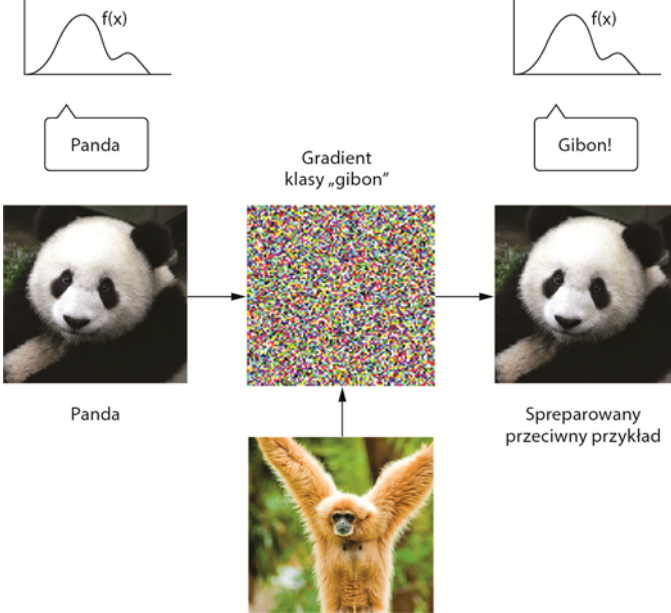


**Rysunek 8.17.** Artefakty przypominające swym wyglądem szachownice; są wywołane przez brak korelacji między rozmiarem kroku a rozmiarem jądra w pokryciu przestrzeni pikseli; jest to jeden z wielu problemów spotykanych podczas pracy z sieciami GAN

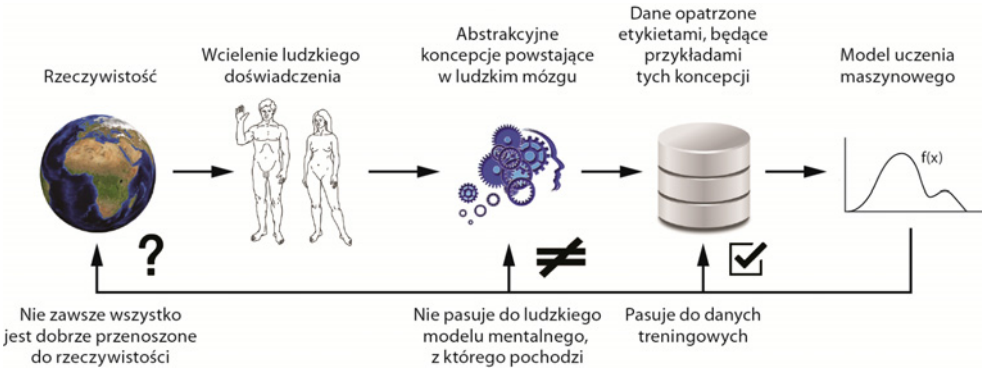


**Rysunek 8.18.** Pobaw się w dyskrminator: w każdej kolumnie znajdują się dwa obrazy wygenerowane przez sieć GAN i jeden obraz pochodzący z treningowego zbioru danych. Czy możesz odróżnić je od siebie? (Poprawne odpowiedzi: prawdziwe obrazy w kolejnych kolumnach umieszczono na środku, u góry, u dołu i na środku)

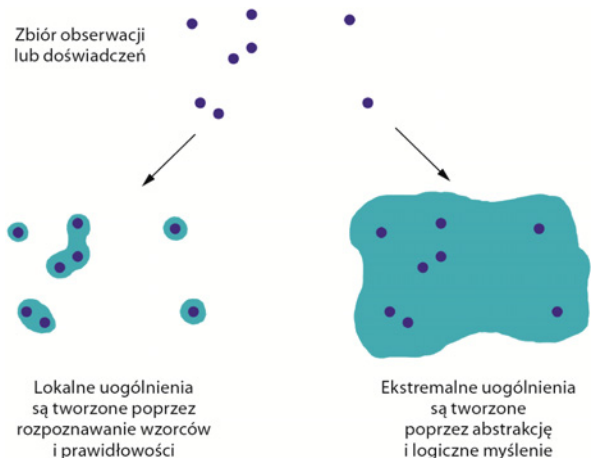
# Rozdział 9.



**Rysunek 9.2.** Przykład przeciwnego obrazu zmodyfikowanego tak, aby został zaklasyfikowany przez model w innej klasie

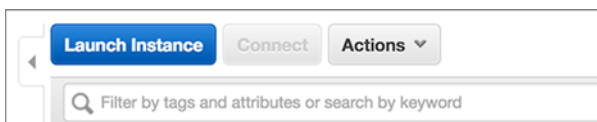


**Rysunek 9.3.** Obecne modele uczenia maszynowego przypominają zamglone lustrzane odbicia rzeczywistych koncepcji

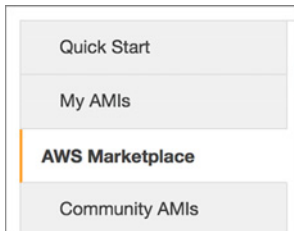


**Rysunek 9.4.** Porównanie mechanizmów lokalnego i ekstremalnego uogólniania

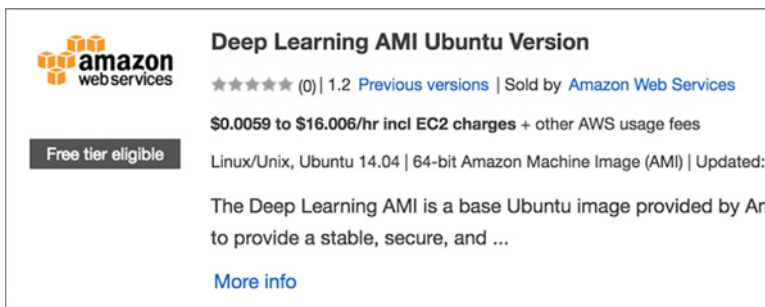
## Dodatek B



**Rysunek B.1.** Panel sterowania usługi EC2



**Rysunek B.2.** Otwieranie sklepu instancji EC2



**Rysunek B.3.** Obraz maszyny przeznaczonej do uczenia głębokiego pracującej pod kontrolą systemu Ubuntu

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

Step 2: Choose an Instance Type

<input type="checkbox"/>	GPU instances	g2.8xlarge	32
<input checked="" type="checkbox"/>	GPU compute	p2.xlarge	4
<input type="checkbox"/>	GPU compute	p2.8xlarge	32

Rysunek B.4.  
Instancja p2.xlarge

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group  
☐ Select an existing security group

Security group name:

Description: This security group was generated by AWS Marketplace and is based on recom

Type	Protocol	Port Range	Source
SSH	TCP	22	Custom 0.0.0.0/0
Custom TCP Rule	TCP	8888	Anywhere 0.0.0.0/0, ::0

Add Rule

Rysunek B.5. Konfiguracja zabezpieczeń

Connect To Your Instance

I would like to connect with

☒ A standalone SSH client

☐ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))

2. Locate your private key file (awsKeys.pem). The wizard automatically detects the key you used to launch the instance.

3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 awsKeys.pem
```

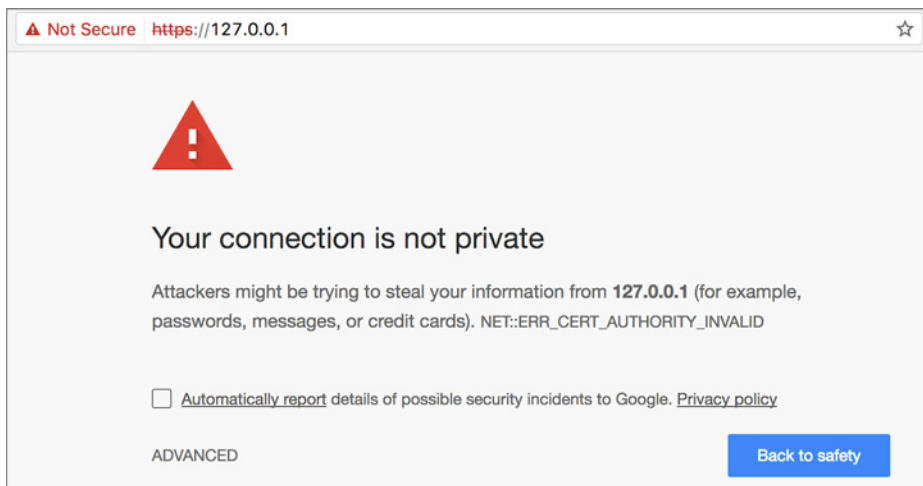
4. Connect to your instance using its Public DNS:

```
ec2-54-147-126-214.compute-1.amazonaws.com
```

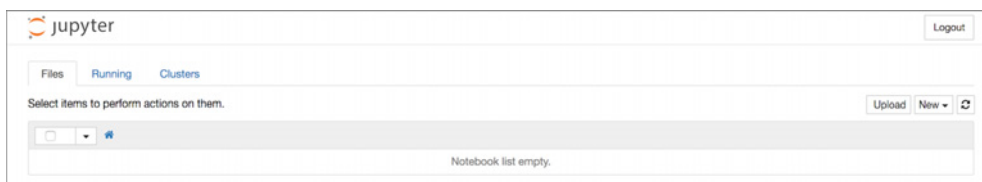
Example:

```
ssh -i "awsKeys.pem" ubuntu@ec2-54-147-126-214.compute-1.amazonaws.com
```

Rysunek B.6.  
Instrukcja łączenia się z instancją



**Rysunek B.7.** Ostrzeżenie, które można zignorować



**Rysunek B.8.** Ekran główny środowiska Jupyter