Interpretability of Time Series Analysis with DeepLearning

Overview and Tutorial

Jörg Simon, 30.April 2020

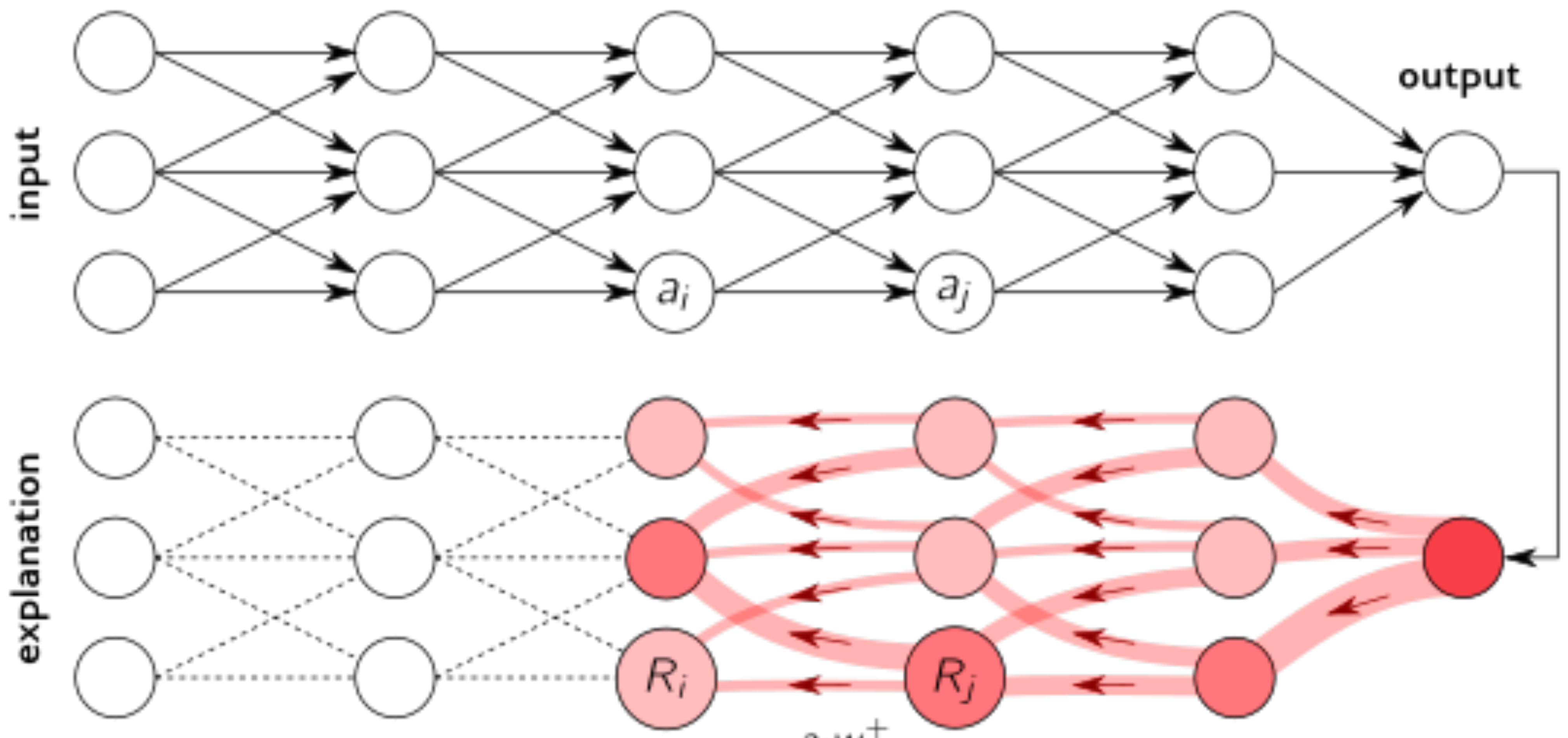
About me

- PhD on using DeepLearning to detect Human Factors from BioSignals
- Prof. Eduardo Veas and Herbert Danzinger
- Sometimes very Sparse Data!
- Inspired to use interpretability results to change the training process itself.

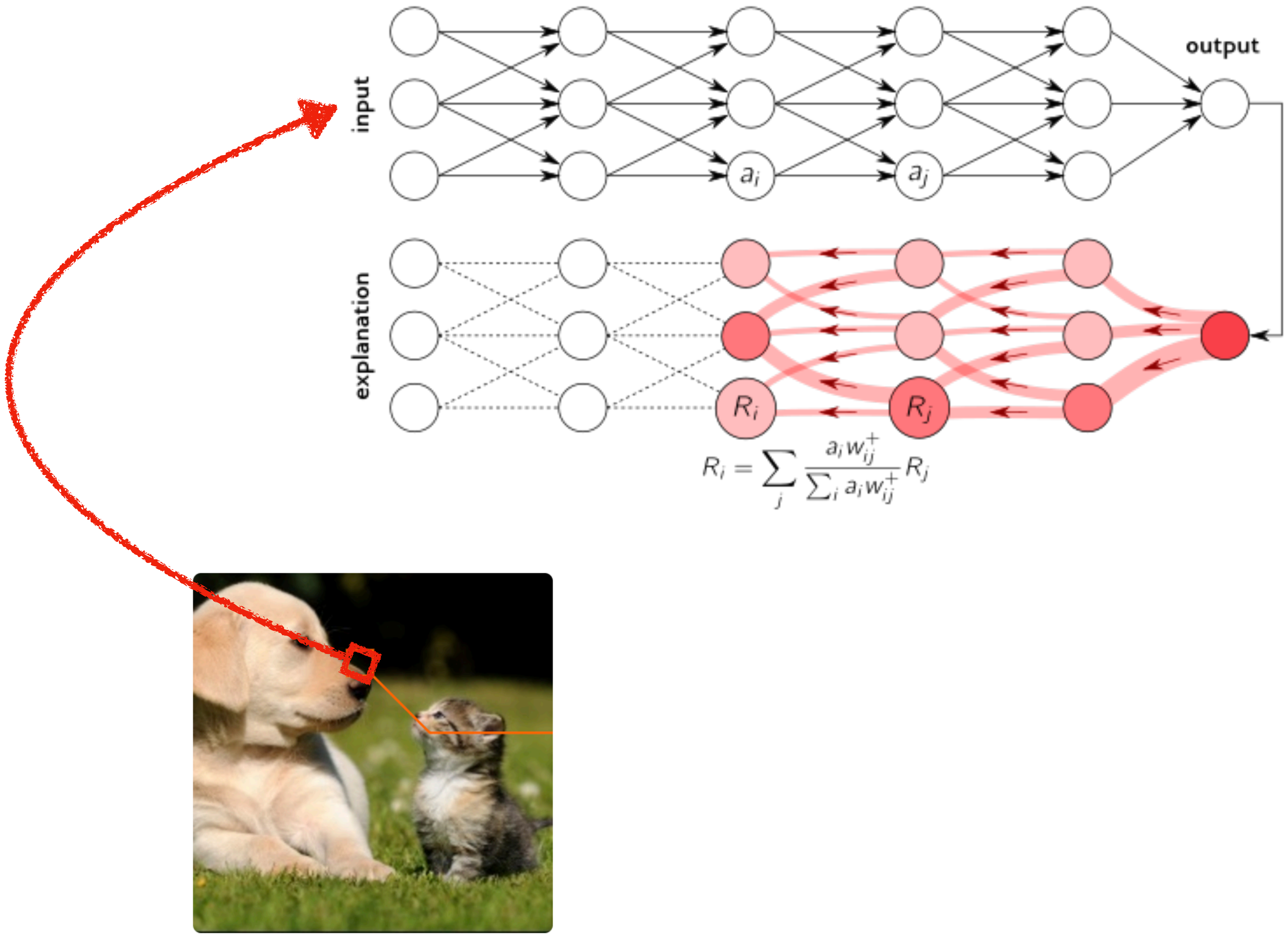
Ressources

- https://github.com/grazai/xai-tutorial-april-2020
- http://projector.tensorflow.org/
- https://distill.pub/2016/misread-tsne/
- Karpathy et al. 2015: https://arxiv.org/pdf/1506.02078.pdf
- https://distill.pub/2019/memorization-in-rnns/
- https://github.com/HendrikStrobelt/LSTMVis

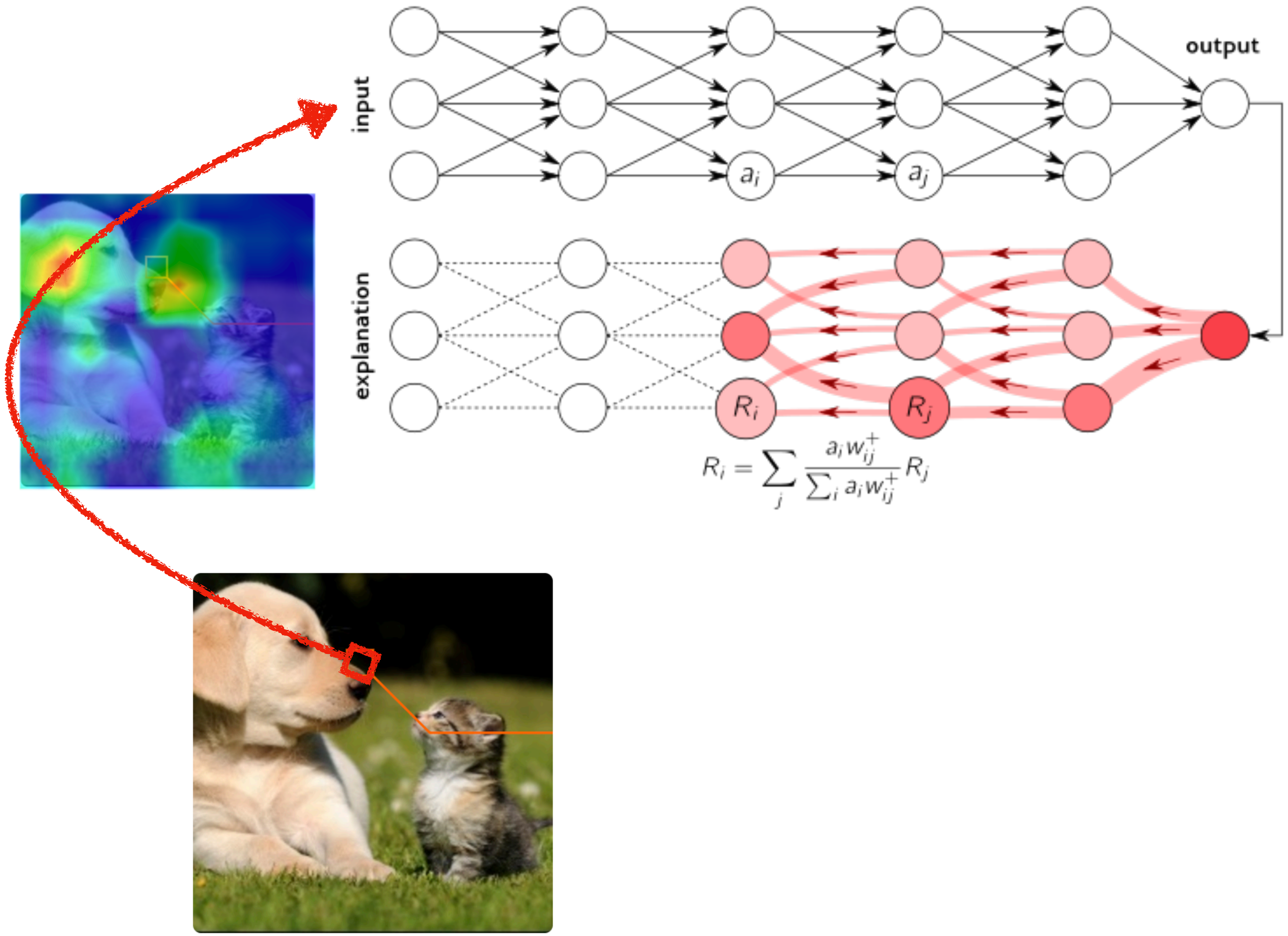
Small Recap

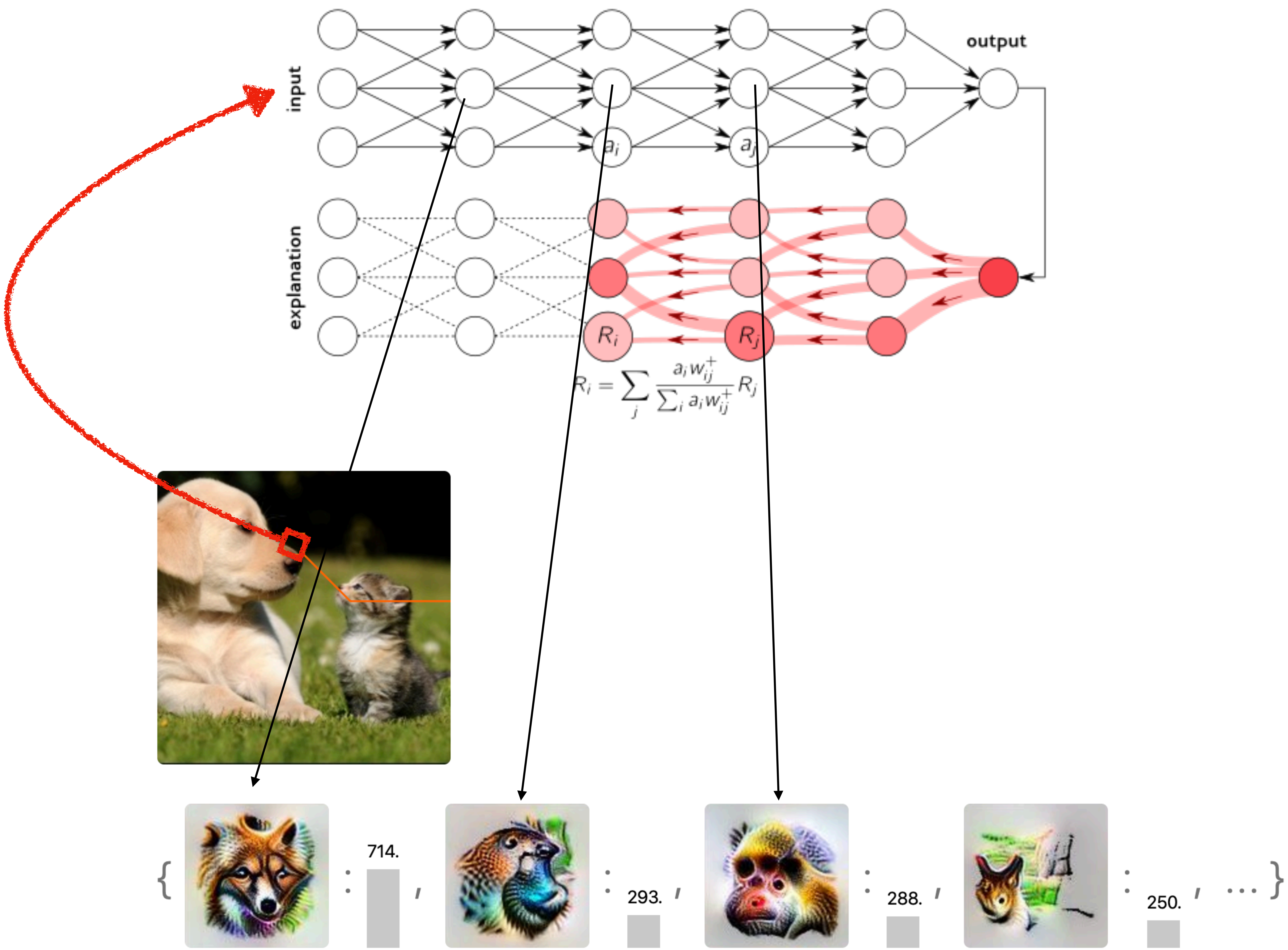


$$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$$

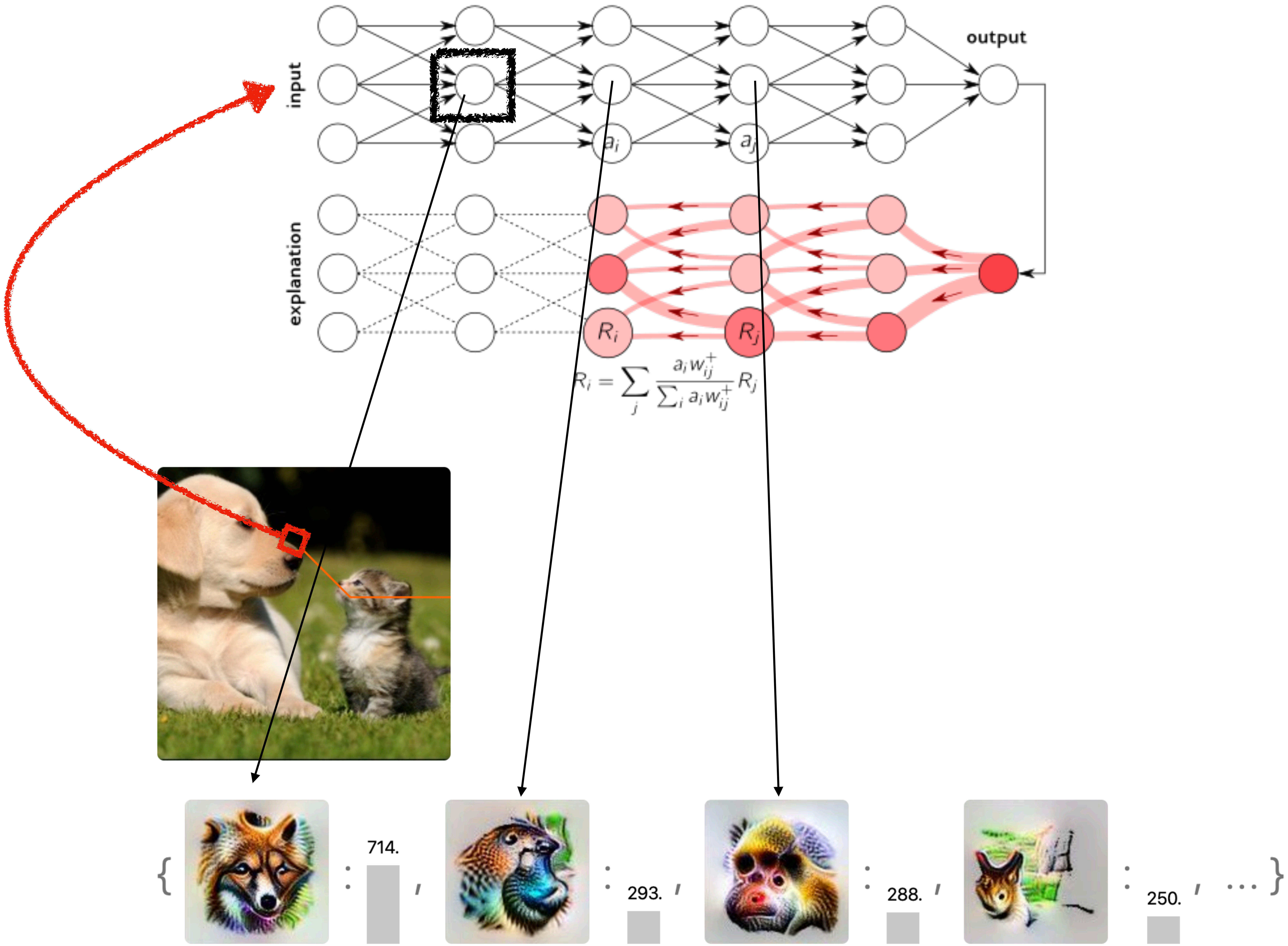


input
output
explanation
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$

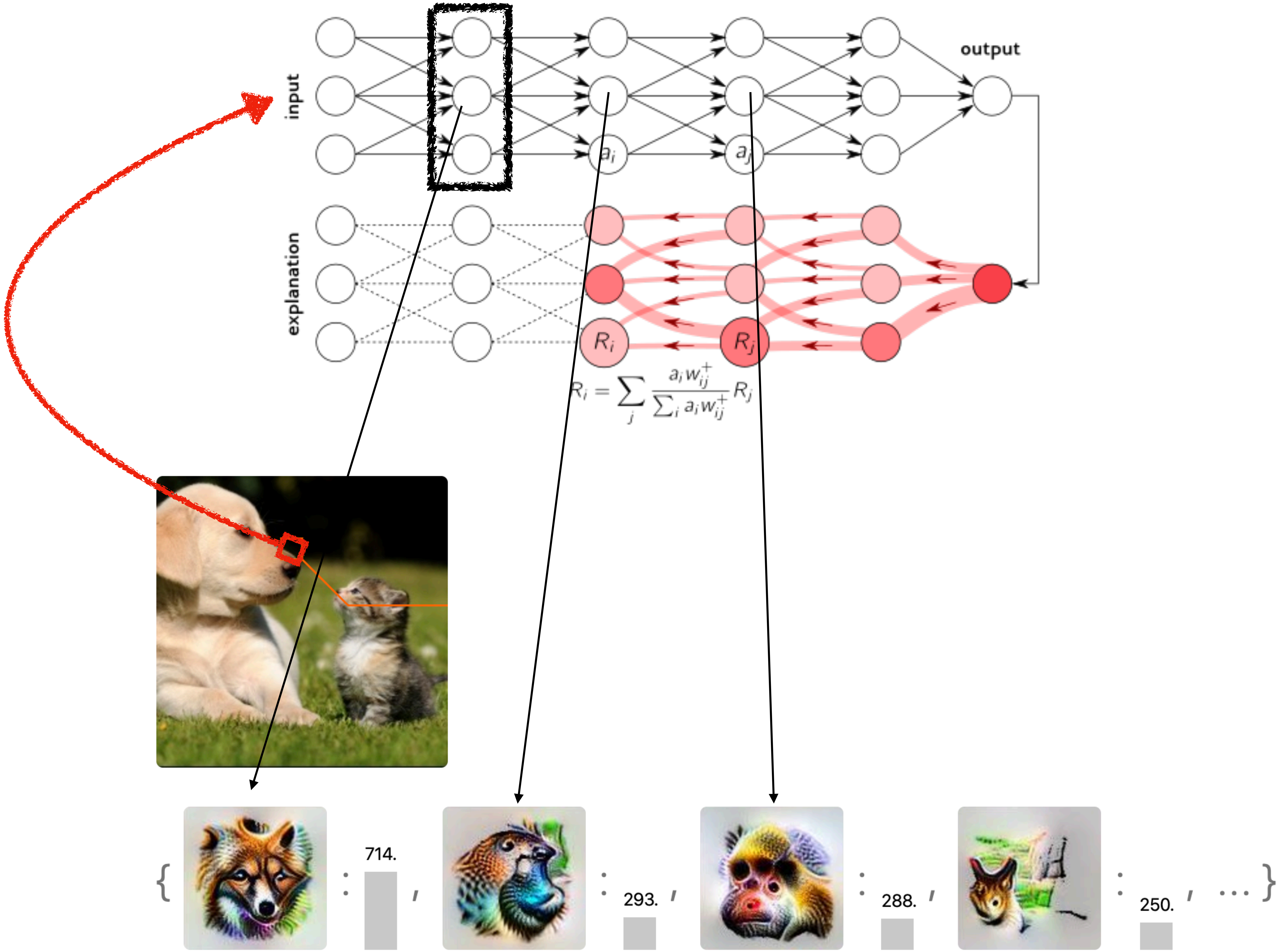

input
output
explanation
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$



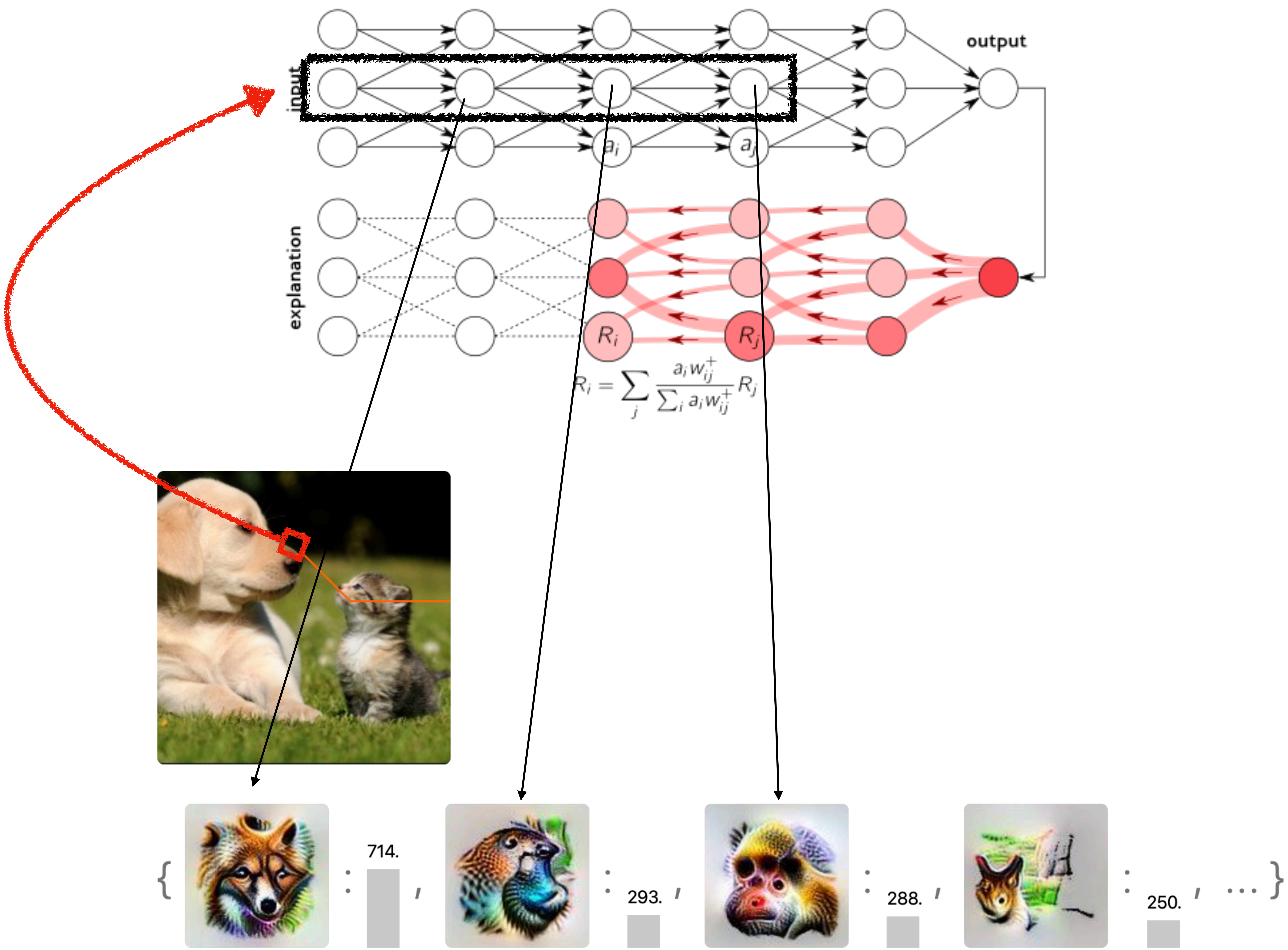
input
output
explanation
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$
714.
293.
288.
250.



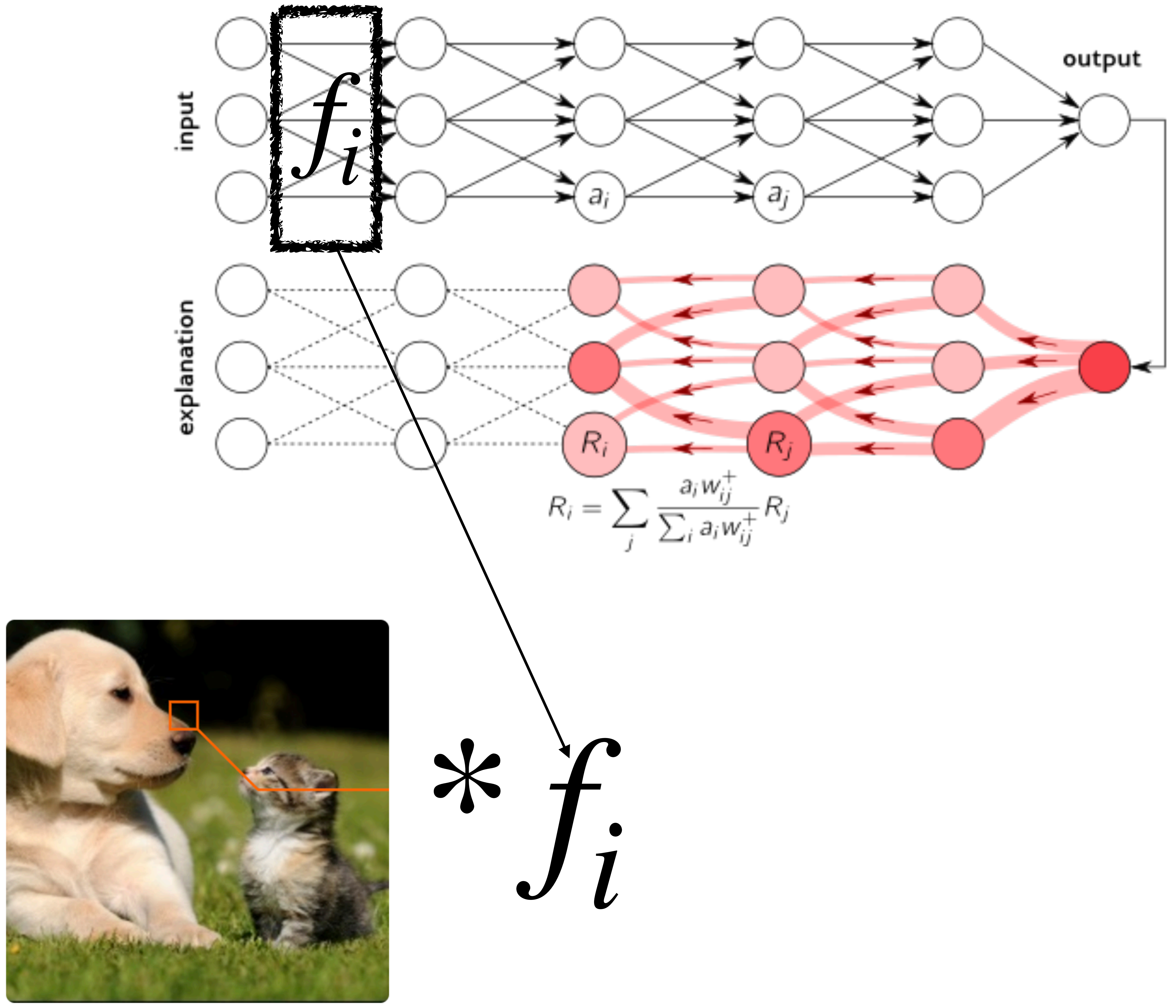
input
output
explanation
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$
714.
293.
288.
250.



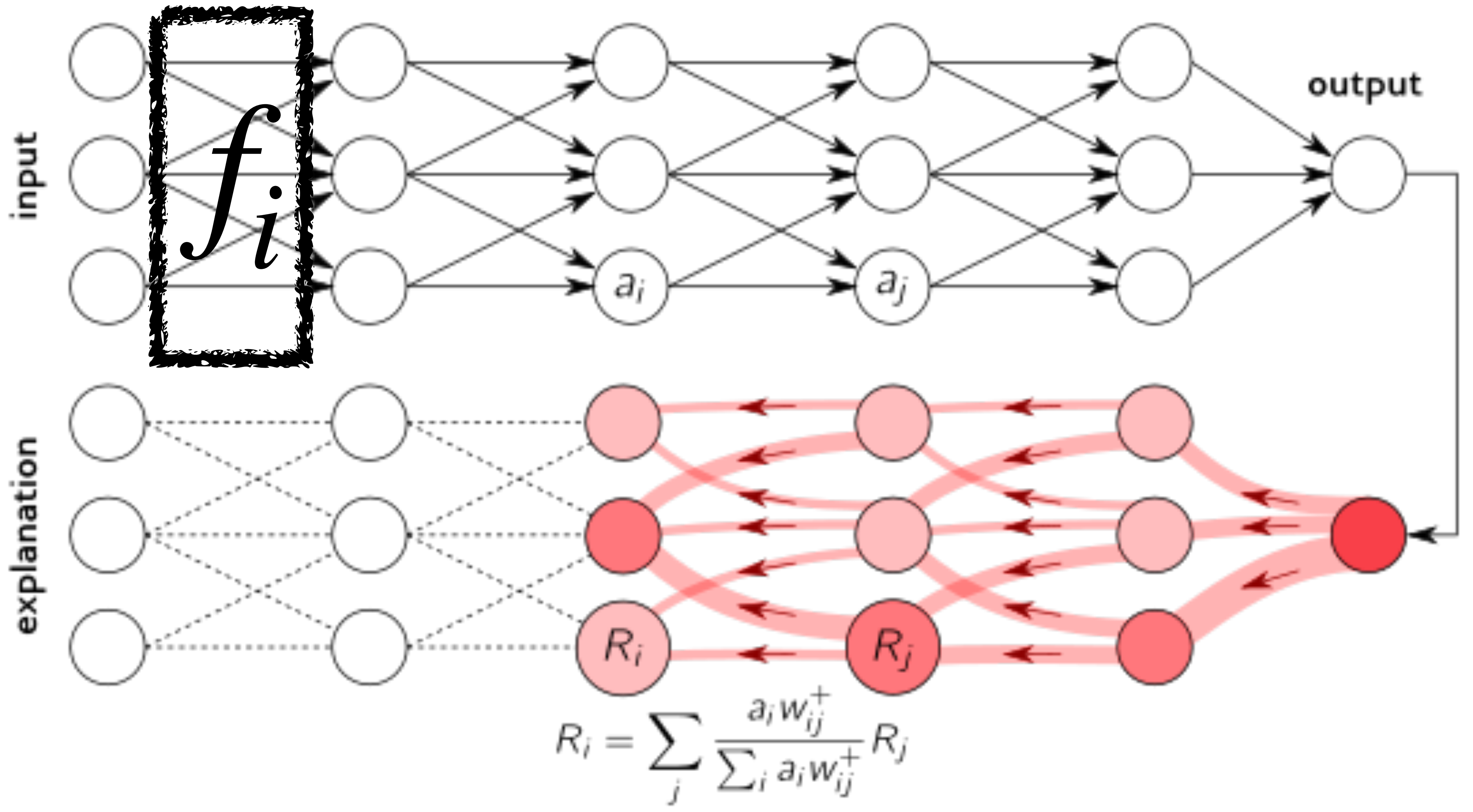
input
output
explanation
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$
{
: 714. ,
: 293. ,
: 288. ,
: 250. , … }

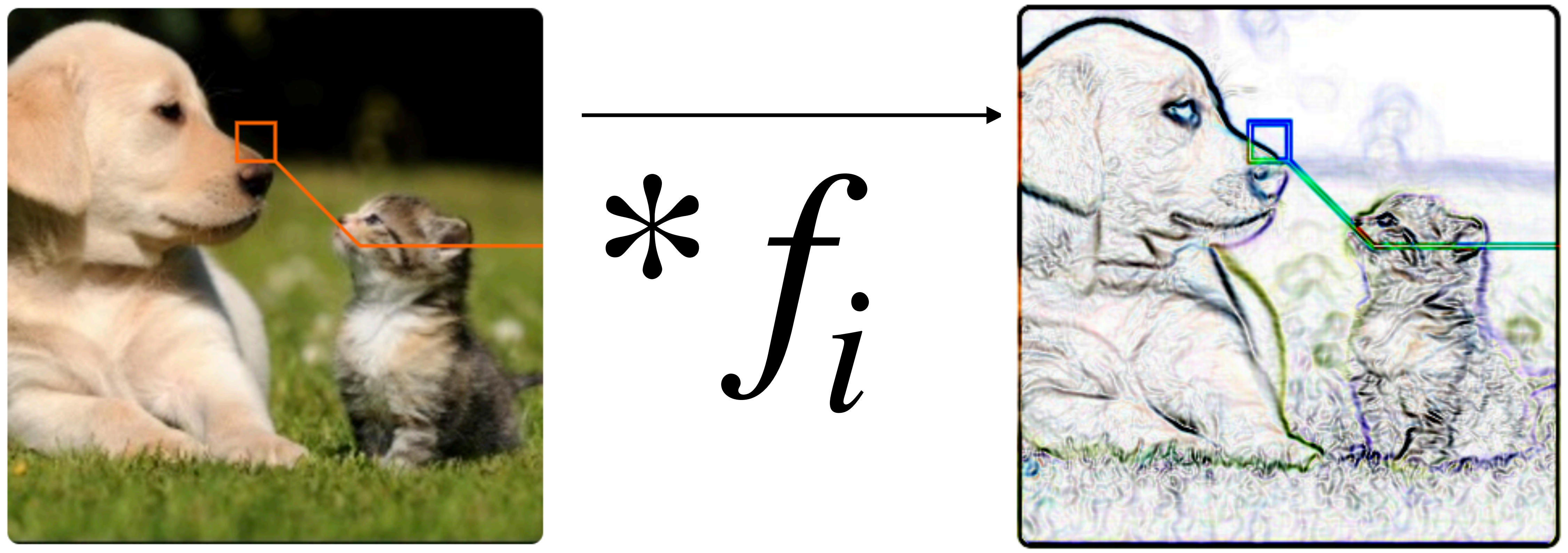


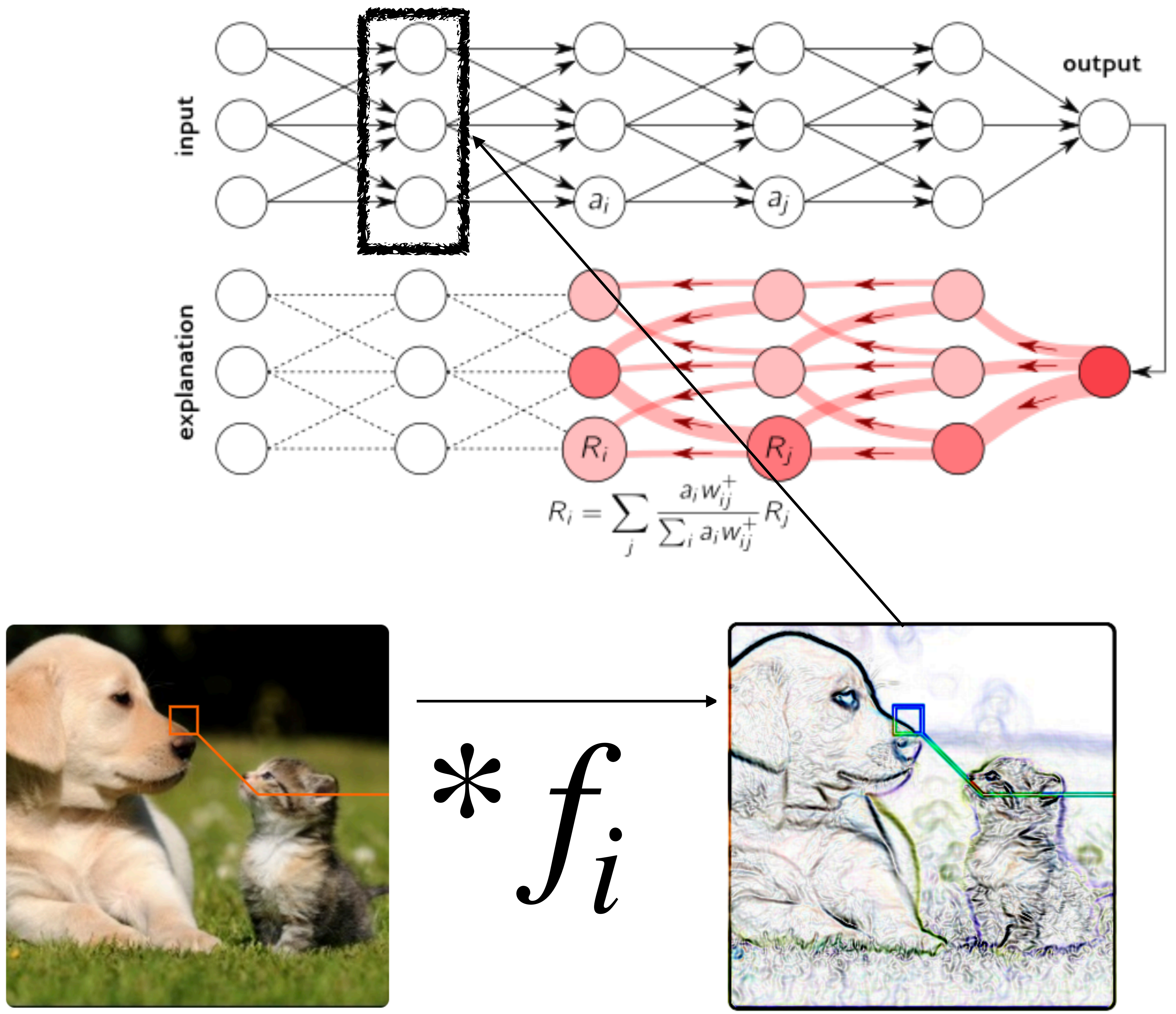
output
input
explanation
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$
{
: 714. ,
: 293. ,
: 288. ,
: 250. , ... }



f_i
input
output
explanation
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$
$* f_i$

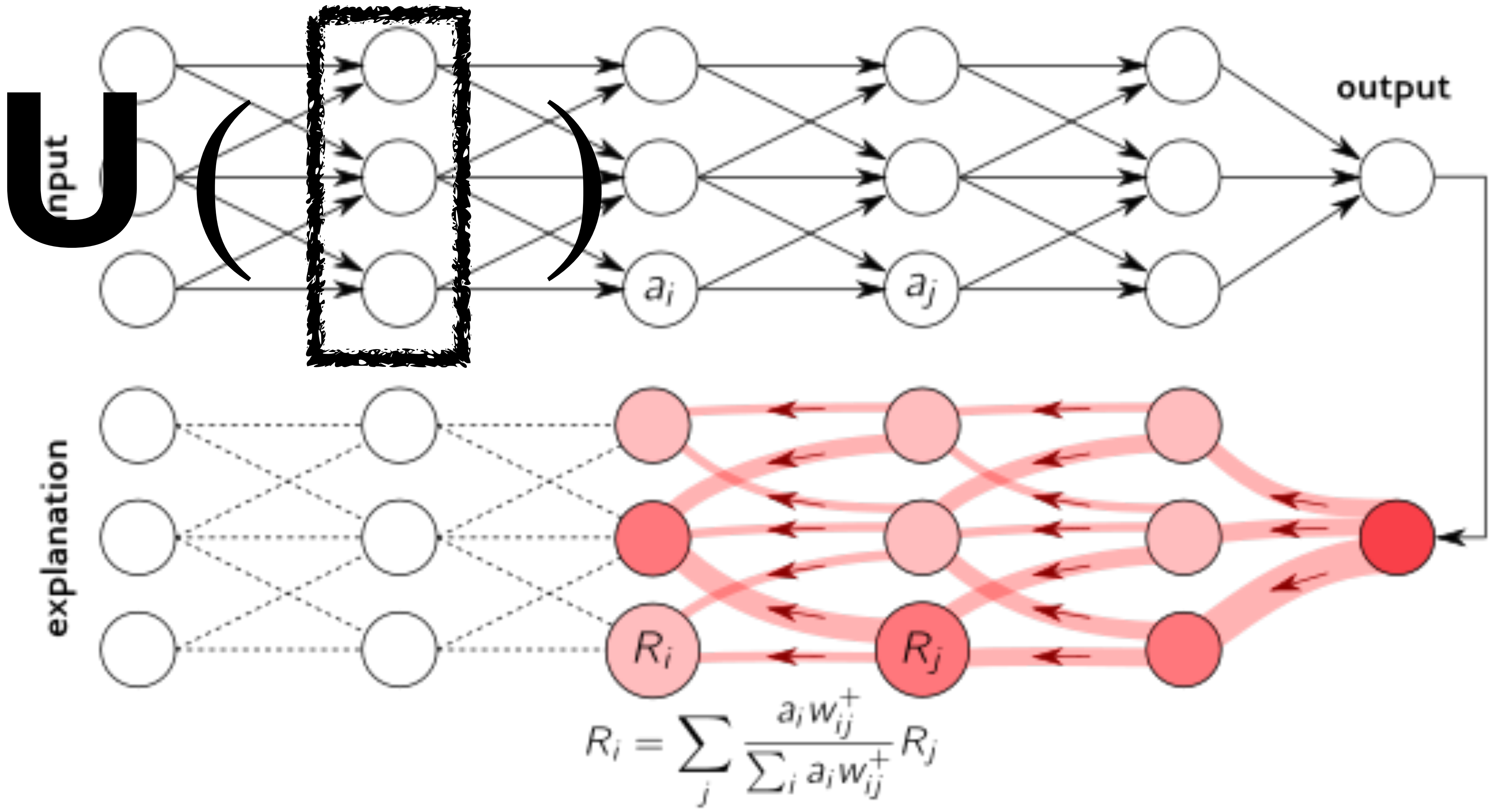

input
output
explanation
f_i
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$


$* f_i$

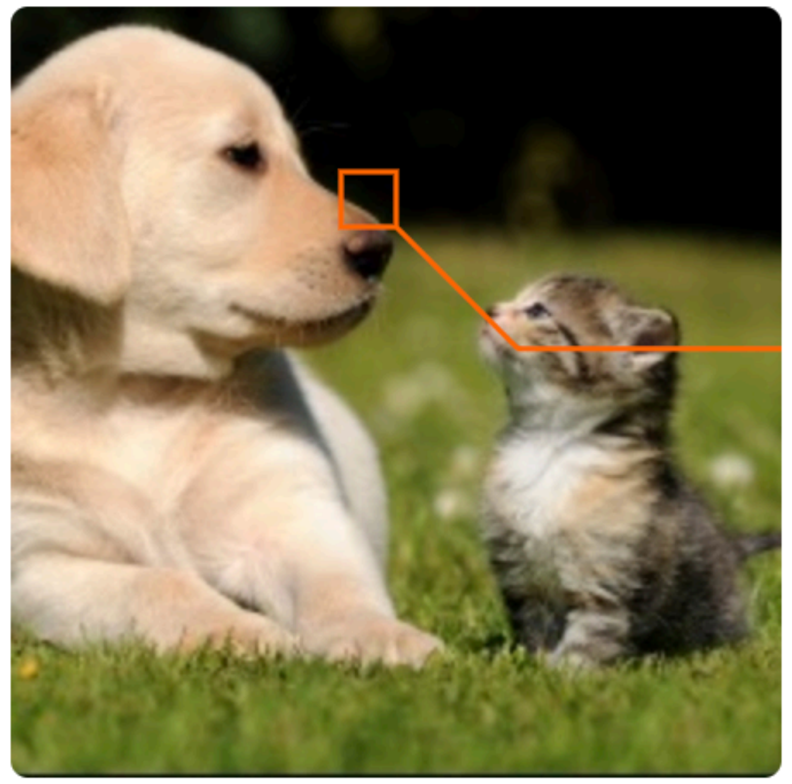


input
output
explanation
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$
$* f_i$

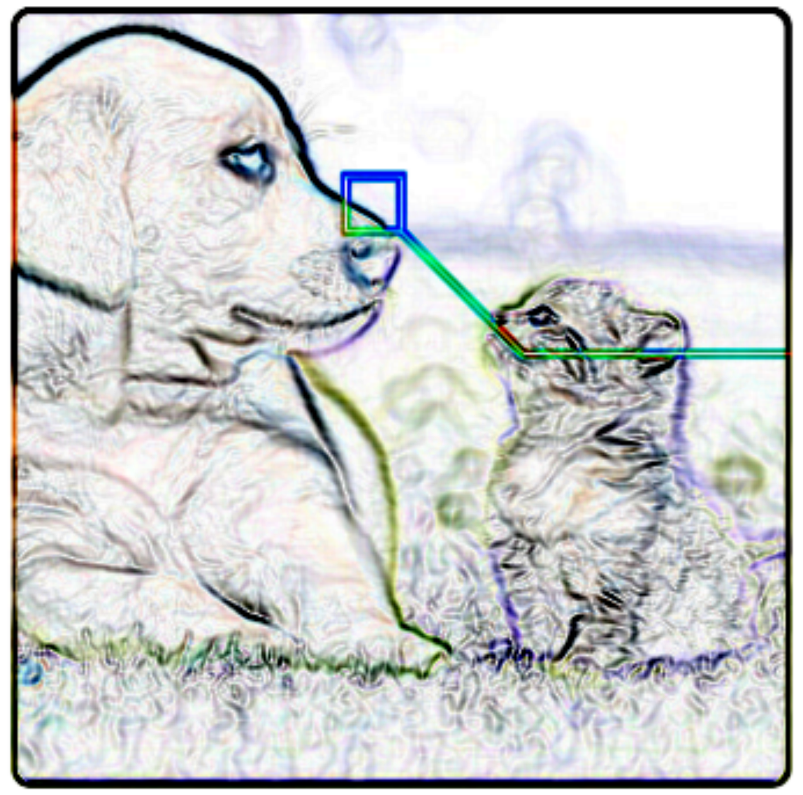
RELU( )

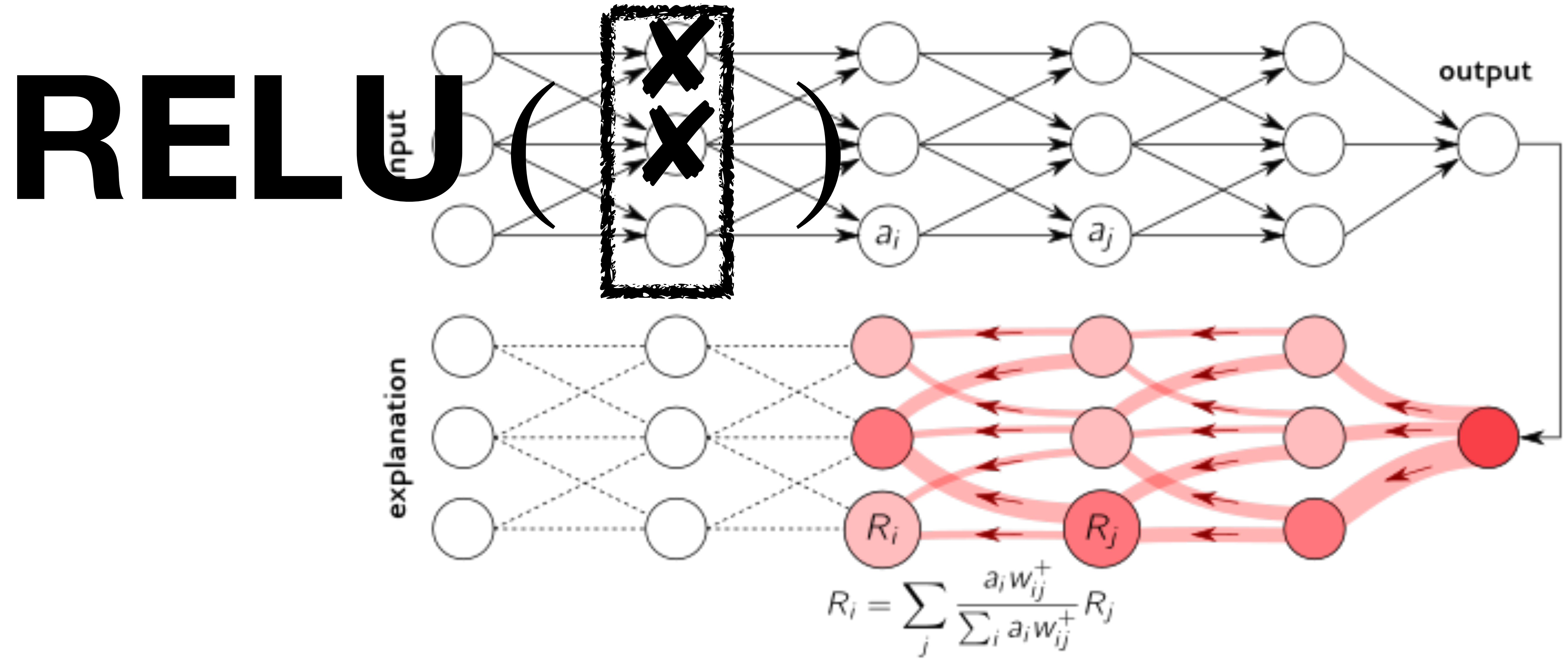


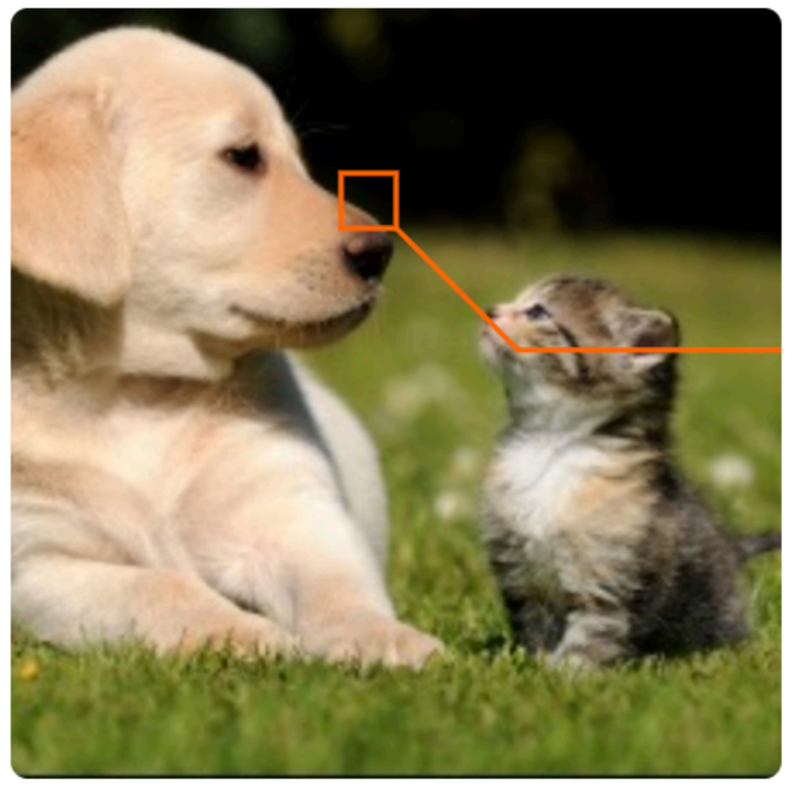
output
nput
explanation
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$



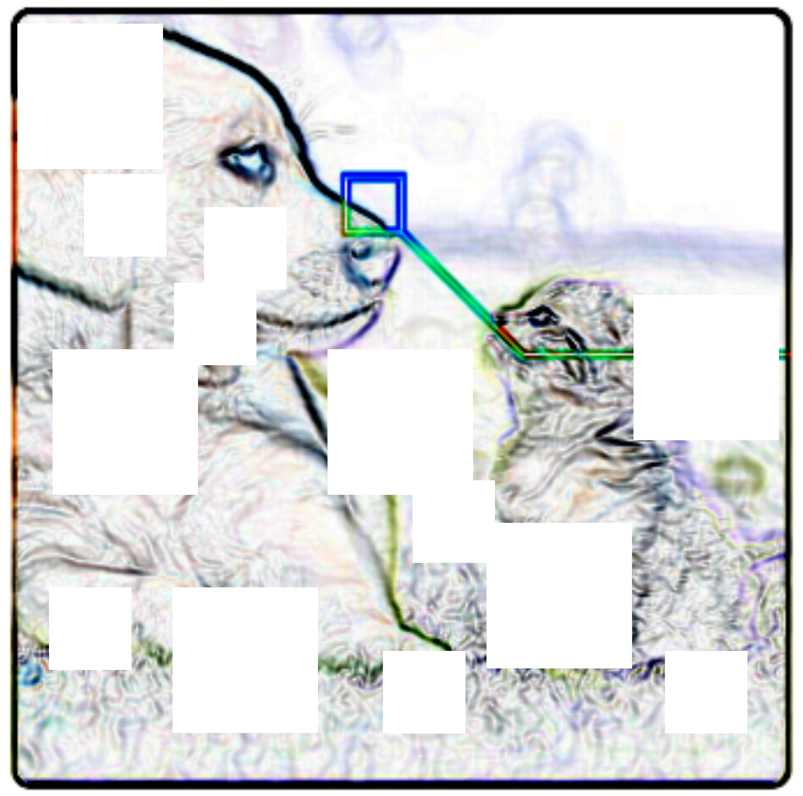
$* f_i$

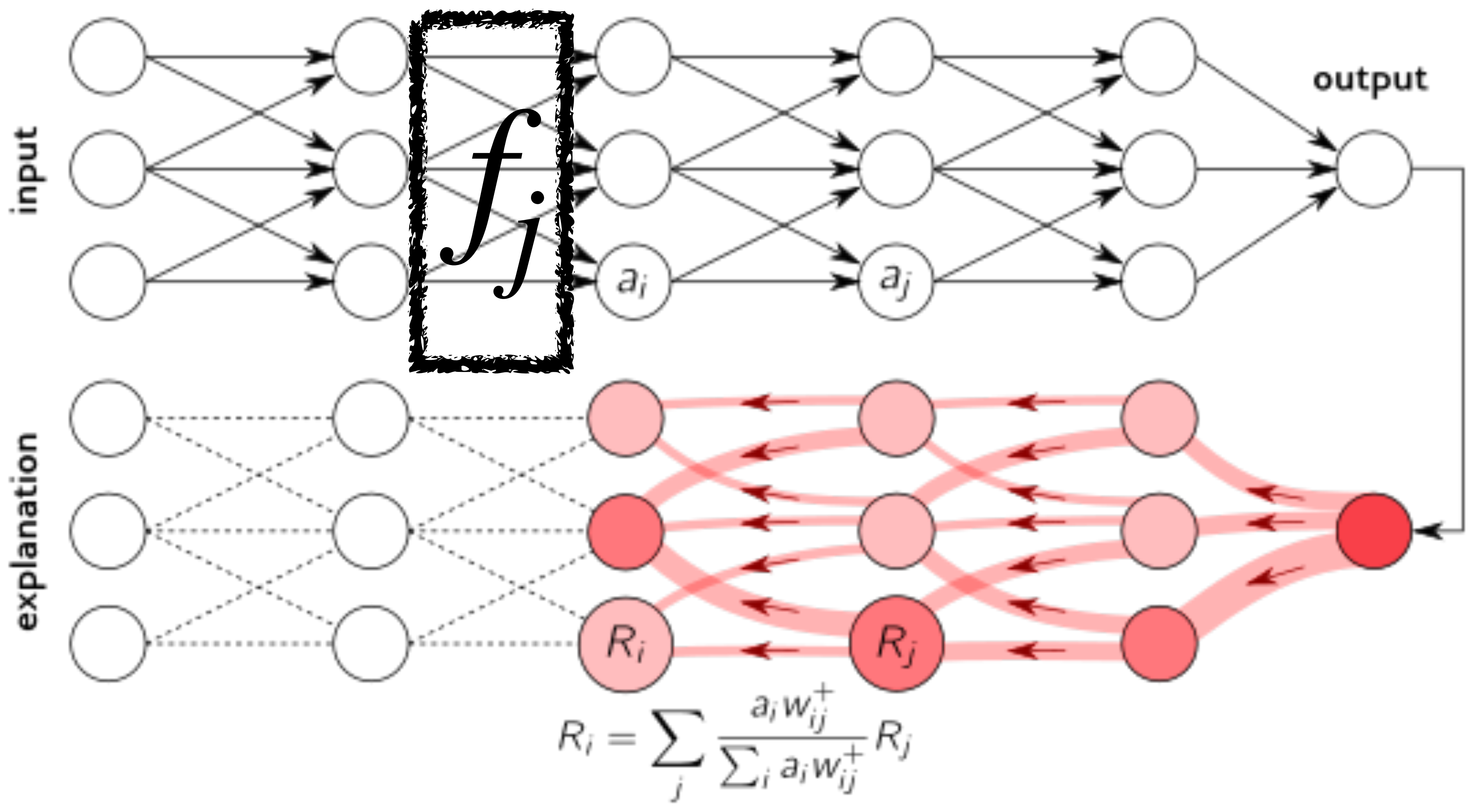


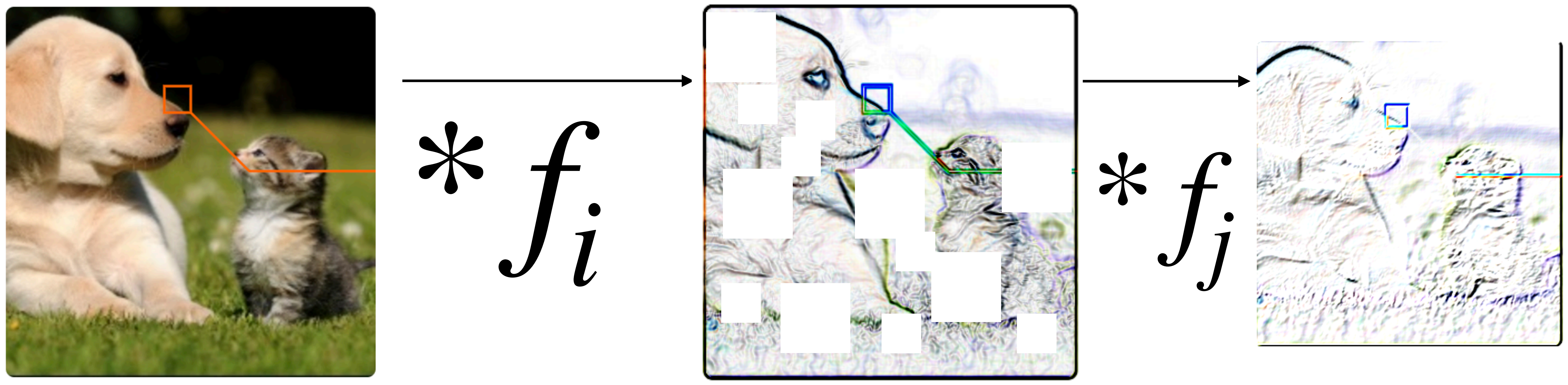

RELU ( )
input
output
explanation
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$

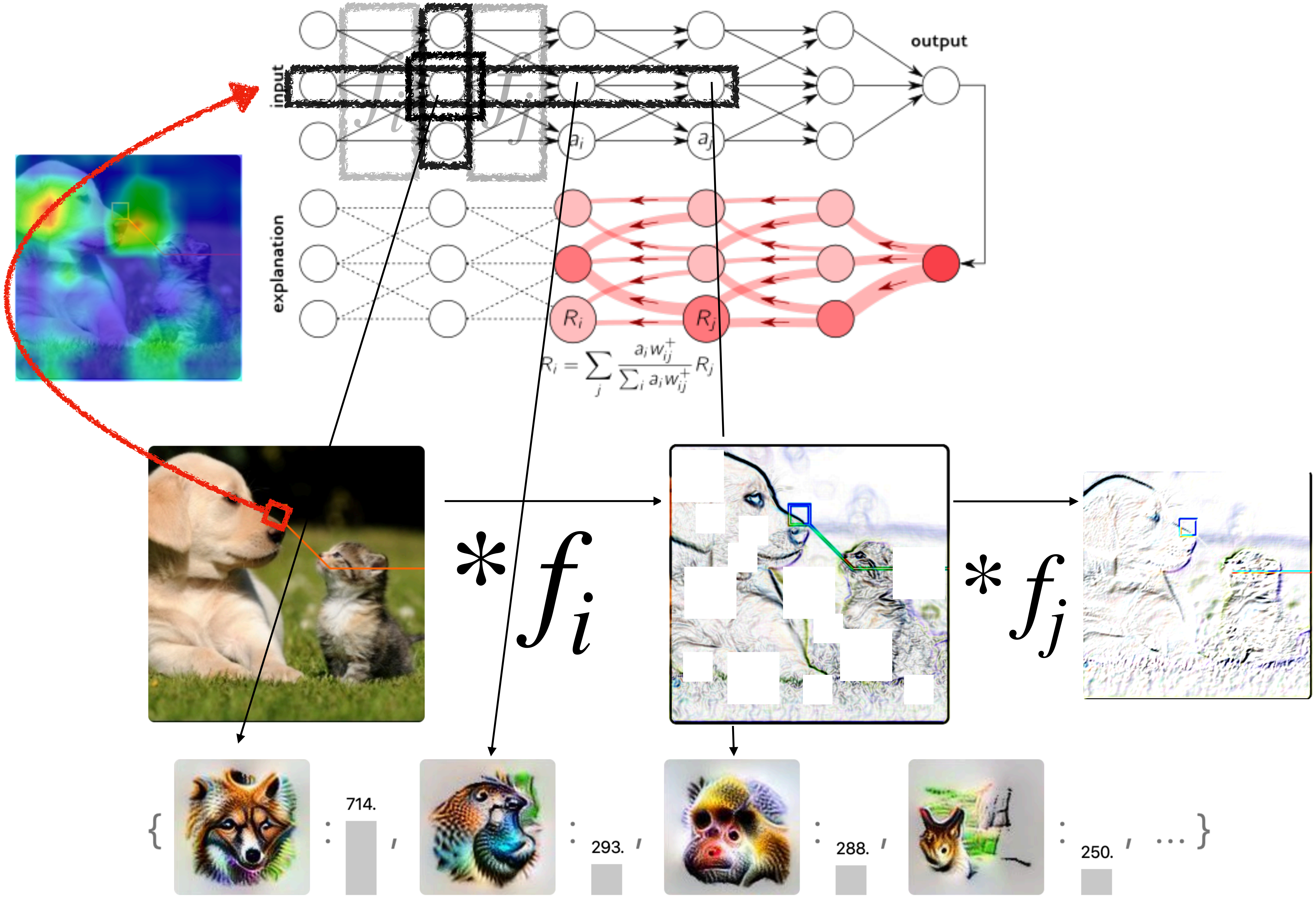


$*f_i$




input
output
explanation
f_j
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$


$*f_i$
$*f_j$



input
output
explanation
a_i
a_j
R_i
R_j
$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$
$* f_i$
$* f_j$
714.
293.
288.
250.

Time Series

What is that?

Time Series

What is that?

- Audio
  - Fixed sampling frequency + start point
  - Date, 20Hz, $[0.532, 1.103, 0.765, 0.111, 0.998]$
  - Spectogram: time bins

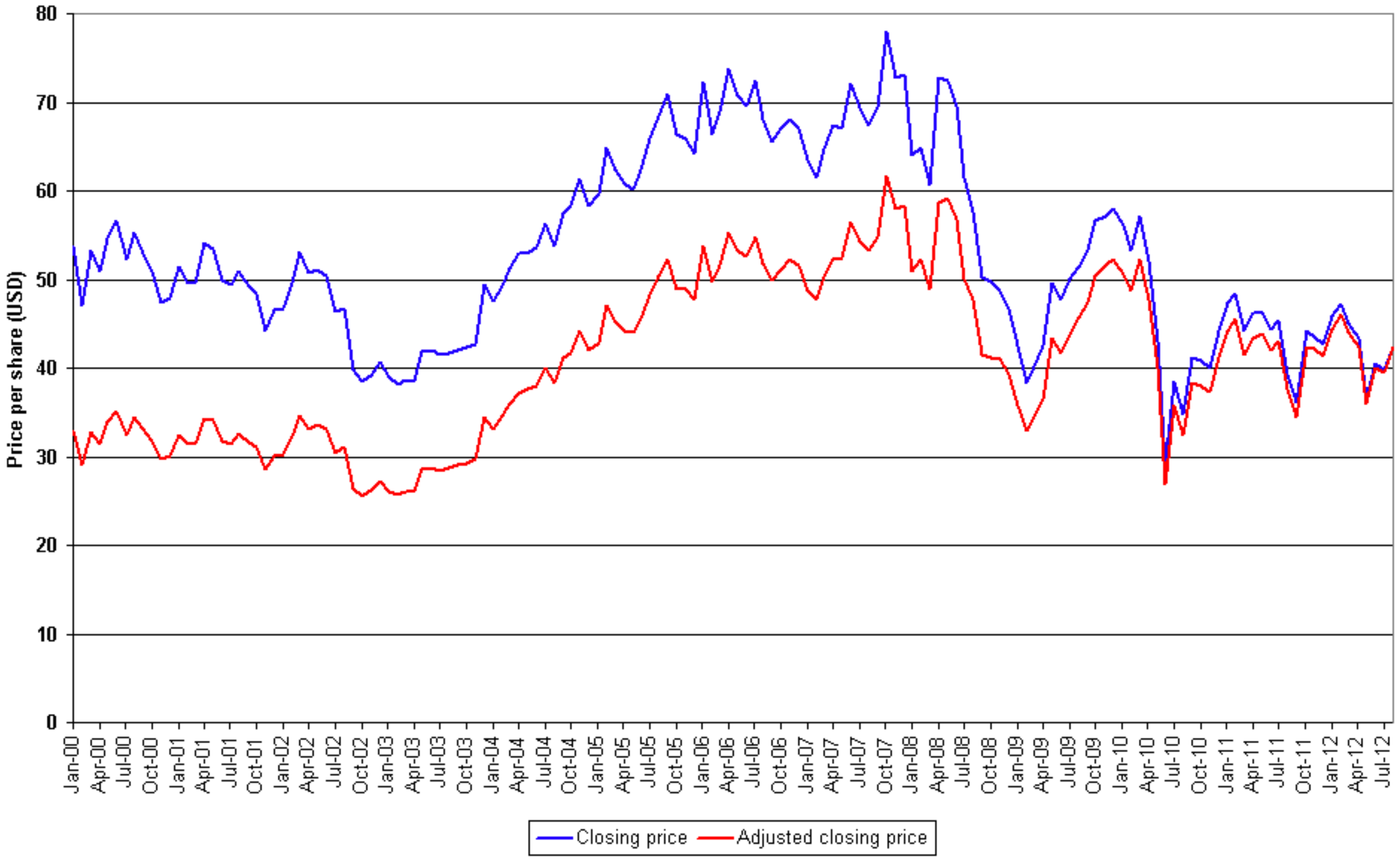
Parallel Bars

(a)

(b)

Time Series
What is that?

- Audio
- Video
  - Audio +
  - 4D Tensor: [Timesteps, (Color-)Channels, Width, Height]

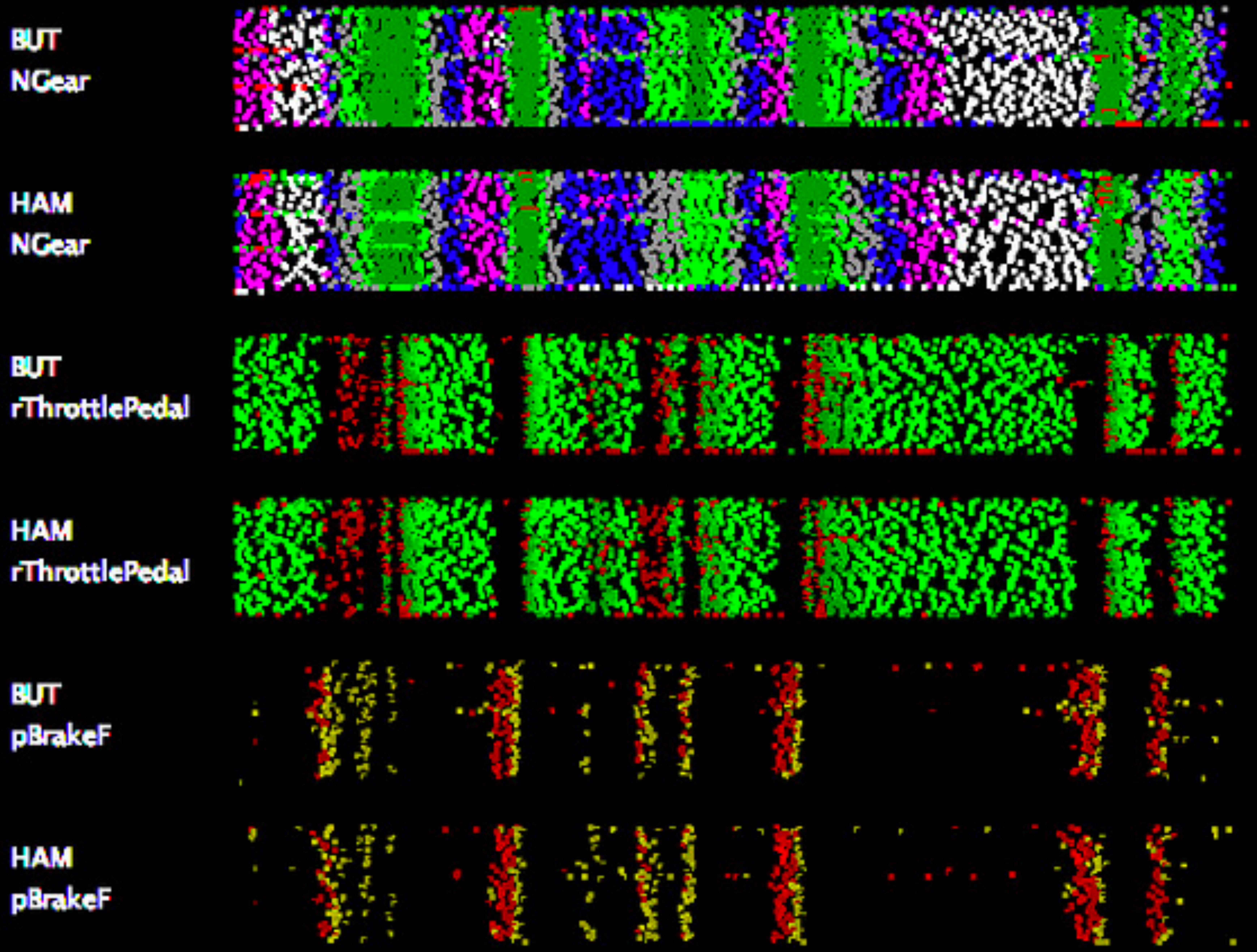

80
70
60
50
40
30
20
10
0
Price per share (USD)
Jan-00 Apr-00 Jul-00 Oct-00 Jan-01 Apr-01 Jul-01 Oct-01 Jan-02 Apr-02 Jul-02 Oct-02 Jan-03 Apr-03 Jul-03 Oct-03 Jan-04 Apr-04 Jul-04 Oct-04 Jan-05 Apr-05 Jul-05 Oct-05 Jan-06 Apr-06 Jul-06 Oct-06 Jan-07 Apr-07 Jul-07 Oct-07 Jan-08 Apr-08 Jul-08 Oct-08 Jan-09 Apr-09 Jul-09 Oct-09 Jan-10 Apr-10 Jul-10 Oct-10 Jan-11 Apr-11 Jul-11 Oct-11 Jan-12 Apr-12 Jul-12
Closing price
Adjusted closing price


35,933
5.970
539,137
1.720
0.314
1.190
0.332

Time Series

What is that?

- Audio
- Video
- Stock Data
- $M = \begin{bmatrix} 100 & 110 & 82 & 132 & 32 \\ \text{23:10} & \text{23:11} & \text{23:15} & \text{23:16} & \text{23:20} \end{bmatrix}$


BUT
NGear
HAM
NGear
BUT
rThrottlePedal
HAM
rThrottlePedal
BUT
pBrakeF
HAM
pBrakeF

Time Series

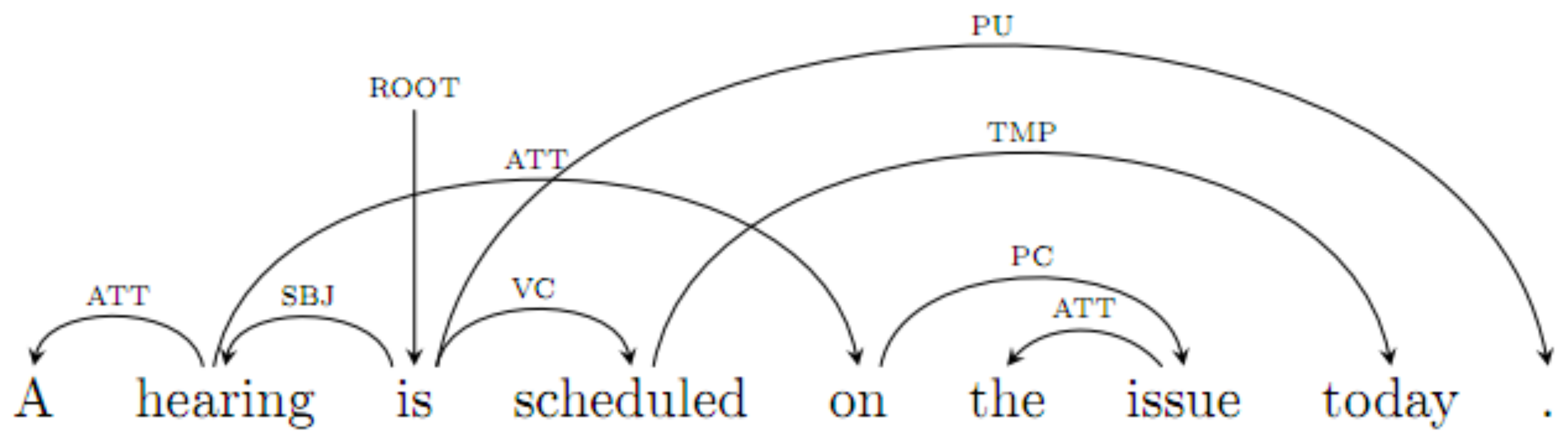
What is that?

- Audio
- Video
- Stock Data
- Car Telemetry
- $$M = \begin{bmatrix} 100 & 110 & 82 & 132 & 32 \\ 10 & 20 & 112 & 32 & 2 \\ 300 & 210 & 212 & 13 & 320 \\ 300 & 410 & 382 & 502 & 244 \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$


a
b
c

Time Series
What is that?

- Audio
- Video
- Stock Data
- Car Telemetry
- Speech (= audio with assumed structure)



The sole importance of the crossing of the Berezina lies in the fact that it plainly and indubitably proved the fallacy of all the plans for cutting off the enemy's retreat and the soundness of the only possible line of action--the one Kutuzov and the general mass of the army demanded--namely, simply to follow the enemy up. The French crowd fled at a continually increasing speed and all its energy was directed to reaching its goal. It fled like a wounded animal and it was impossible to block its path. This was shown not so much by the arrangements it made for crossing as by what took place at the bridges. When the bridges broke down, unarmed soldiers, people from Moscow and women with children who were with the French transport, all--carried on by vis inertiae--pressed forward into boats and into the ice-covered water and did not, surrender.

Cell that turns on inside quotes:

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

Cell that robustly activates inside if statements:

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
    siginfo_t *info)
{
  int sig = next_signal(pending, mask);
  if (sig) {
    if (current->notifier) {
      if (sigismember(current->notifier_mask, sig)) {
        if (!(current->notifier)(current->notifier_data)) {
          clear_thread_flag(TIF_SIGPENDING);
          return 0;
        }
      }
    }
    collect_signal(sig, pending, info);
  }
  return sig;
}
```

A large portion of cells are not easily interpretable. Here is a typical example:

```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
  char *str;
  if (!*bufp || (len == 0) || (len > *remain))
    return ERR_PTR(-EINVAL);
  /* Of the currently implemented string fields, PATH_MAX
   * defines the longest valid length.
   */
```

```
/* Duplicate LSM field information.  The lsm_rule is opaque, so
 * re-initialized. */
static inline int audit_dupe_lsm_field(struct audit_field *df,
          struct audit_field *sf)
{
  int ret = 0;
  char *lsm_str;
  /* our own copy of lsm_str */
  lsm_str = kstrdup(sf->lsm_str, GFP_KERNEL);
  if (unlikely(!lsm_str))
    return -ENOMEM;
  df->lsm_str = lsm_str;
  /* our own (refreshed) copy of lsm_rule */
  ret = security_audit_rule_init(df->type, df->op, df->lsm_str,
              (void **)&df->lsm_rule);
  /* Keep currently invalid fields around in case they
   * become valid after a policy reload. */
  if (ret == -EINVAL) {
    pr_warn("audit rule for LSM \'%s\' is invalid\n",
      df->lsm_str);
    ret = 0;
  }
  return ret;
}
```

Cell that is sensitive to the depth of an expression:

```
#ifdef CONFIG_AUDITSYSCALL
static inline int audit_match_class_bits(int class, u32 *mask)
{
  int i;
  if (classes[class]) {
    for (i = 0; i < AUDIT_BITMASK_SIZE; i++)
      if (mask[i] & classes[class][i])
        return 0;
  }
  return 1;
}
```

Cell that might be helpful in predicting a new line. Note that it only turns on for some ")":

```
char *audit_unpack_string(void **bufp, size_t *remain, si
{
  char *str;
  if (!*bufp || (len == 0) || (len > *remain))
    return ERR_PTR(-EINVAL);
  /* Of the currently implemented string fields, PATH_MAX
   * defines the longest valid length.
   */
  if (len > PATH_MAX)
    return ERR_PTR(-ENAMETOOLONG);
  str = kmalloc(len + 1, GFP_KERNEL);
  if (unlikely(!str))
    return ERR_PTR(-ENOMEM);
  memcpy(str, *bufp, len);
  str[len] = 0;
  *bufp += len;
  *remain -= len;
  return str;
}
```

Time Series

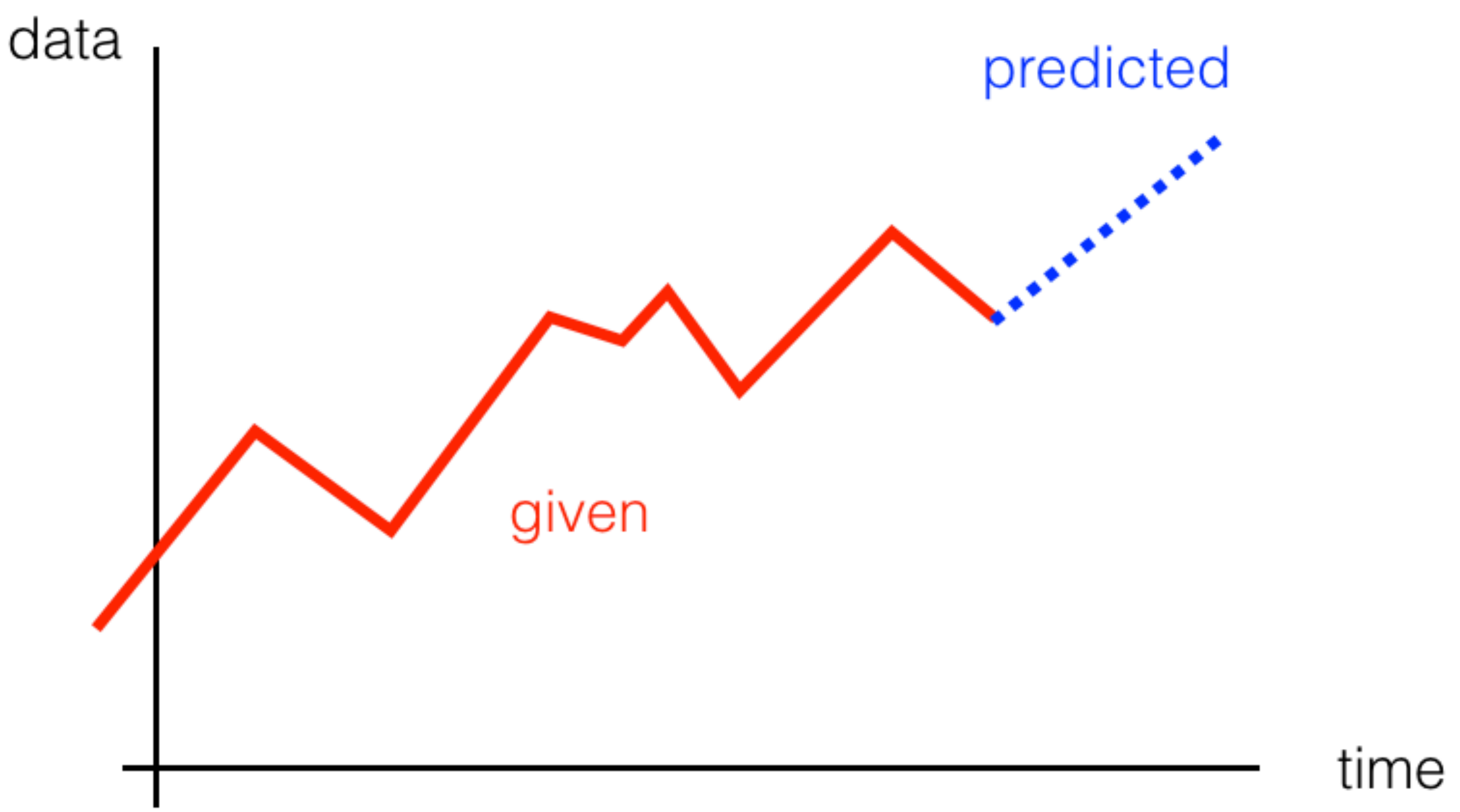
What is that?

- Audio
- Video
- Stock Data
- Car Telemetry
- Speech (= audio with assumed structure)
- Text
- … many more

Time Series

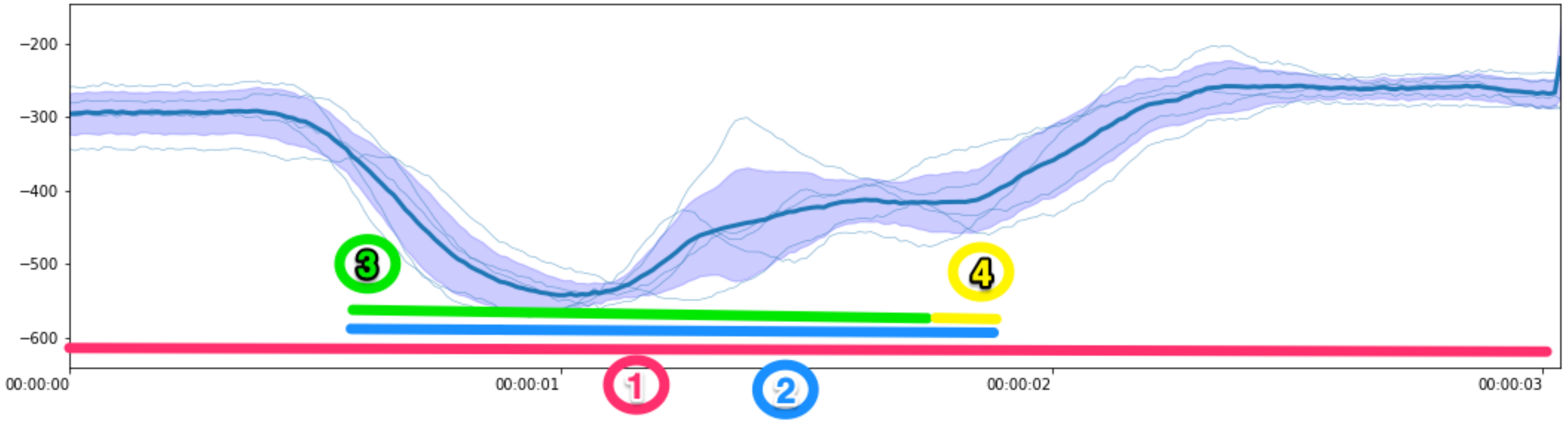
Unit of Analysis

- In Time Series, the unit of analysis is a set of data from the past, and you want to **predict** data from the future
  - Next word in a unfinished sentence
  - Stocks going up or down
  - Steering angle of a car



Time Series

Unit of Analysis



- You might also just reason what a specific Time Series is about (**classification**)
  - Sentiment in a text
  - Emotion encoded within a ECG
  - Parse Tree of a Text

Time Series

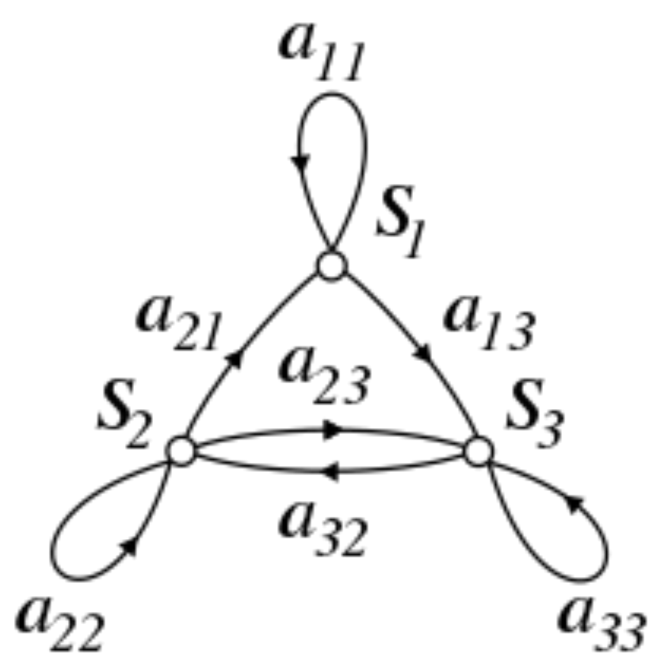
Unit of Analysis

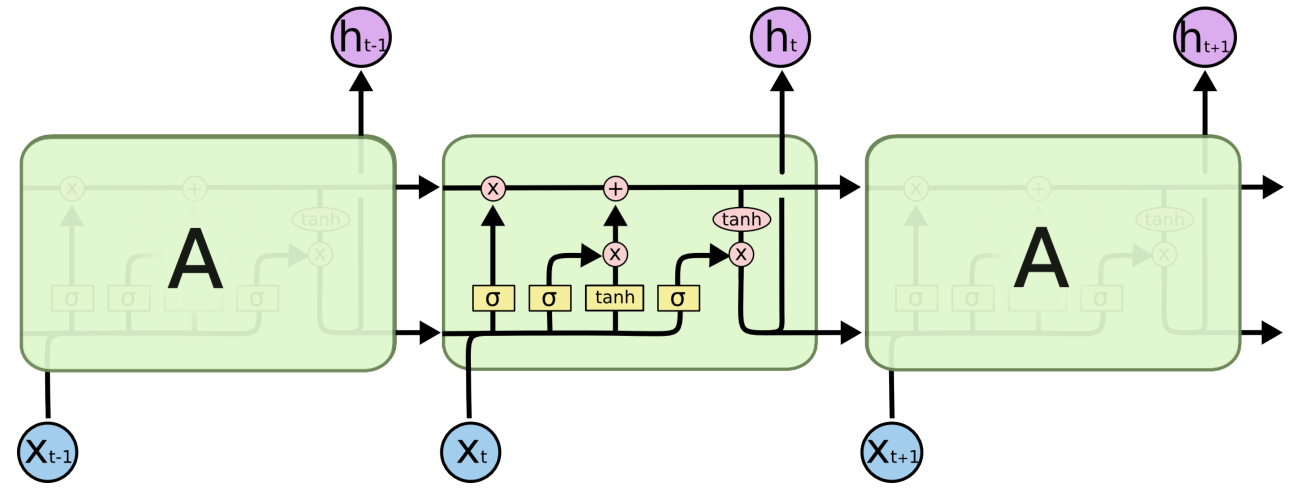
- In Time Series, the unit of analysis is a set of data from the past, and you want to **predict** data from the future
  - Next word in a unfinished sentence
  - Stocks going up or down
  - Steering angle of a car
- You might also just reason what a specific Time Series is about (**classification**)
  - Sentiment in a text
  - Emotion encoded within a ECG
  - Parse Tree of a Text

Time Series

What doe the model model?

- In *all domains* with *all data* usually time series models try to model a **transition** to a new event given the current events
- Markov Models
  - Transition Probability
- DeepLearning: Gates
  - Influence Probability + Morphing
  - (LSTMs, WaveNets, Transformers)





Time Series

What doe the model model?

- In *all domains* with *all data* usually time series models try to model a **transition** to a new event given the current events
- What is an event?
    - Easier (maybe):
        - NLP: Words (Embeddings), Sentences (Biases)
        - Genomes (Embeddings)
    - Harder:
        - General Sensor Data

Time Series

What doe the model model?

- In *all domains* with *all data* usually time series models try to model a **transition** to a new event given the current events
  - Harder:
    - General Sensor Data
    - Single Data Points are taken as Events: This includes all the noise as event data, and is maybe not distinguishable to real events
    - Classical Solutions: Sliding windows or other transformations upfront
    - DeepLearning: Embeddings and Encoders

Time Series

Interpretability

- Two units of Analysis:
  - Semantics of the Encoding of Events
  - Influence of past events for predicting new events

Time Series
Models

- The interpretability methods I talk about are **model dependent.** That means they use specific aspects of the insides of models to allow interpretability
- There are model agnostic methods, I do not talk about them (yet)

Time Series
Models

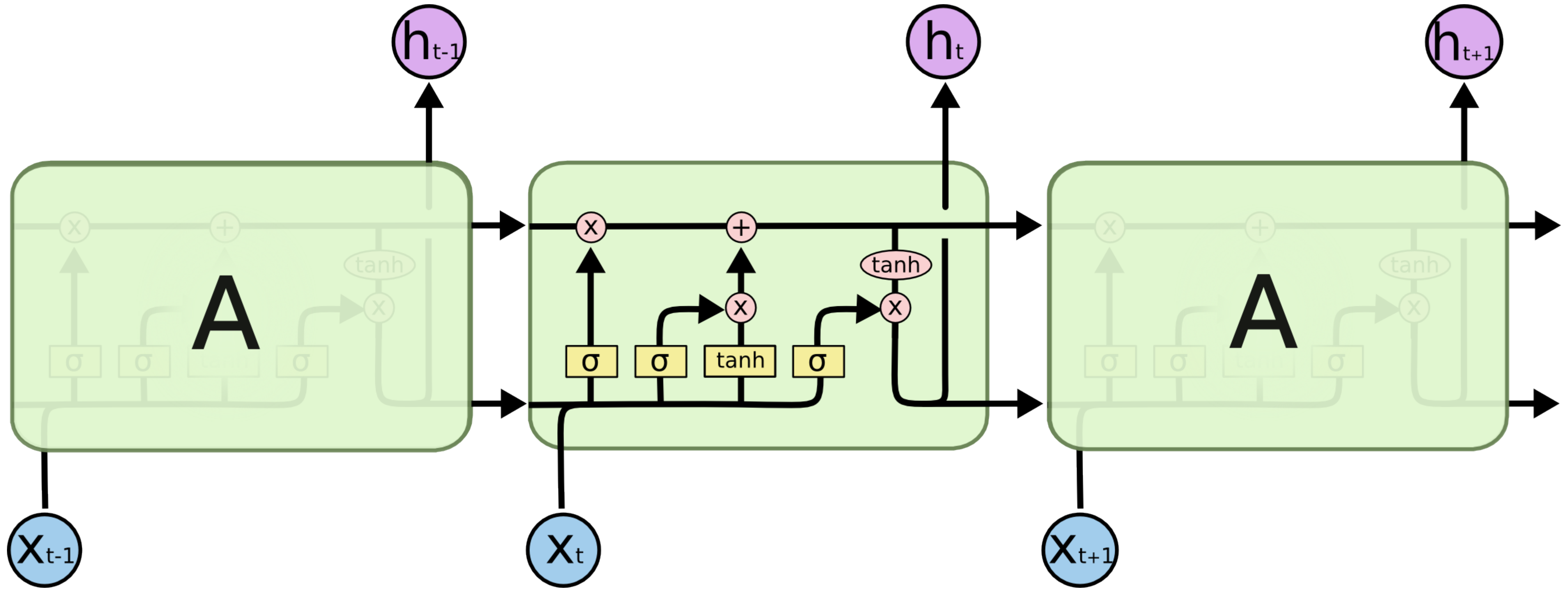
- The interpretability methods I talk about are **model dependent**. That means they use specific aspects of the insides of models to allow interpretability
- RNNs: **LSTMs**, GRUs, …
- Auto-Encoders
- Temporal Convolution Model (TCM), WaveNet (Gated non-overlapping Convolutions)
- Transformers (Multi-Head-Attention Mechanisms {aka. a lot of Gates})
- Hybrids: We also look at LSTM+Encoders in this talk

Time Series

Models

- The interpretability methods I talk about are **model dependent.** That means they use specific aspects of the insides of models to allow interpretability

- RNNs: **LSTMs**, GRUs, …

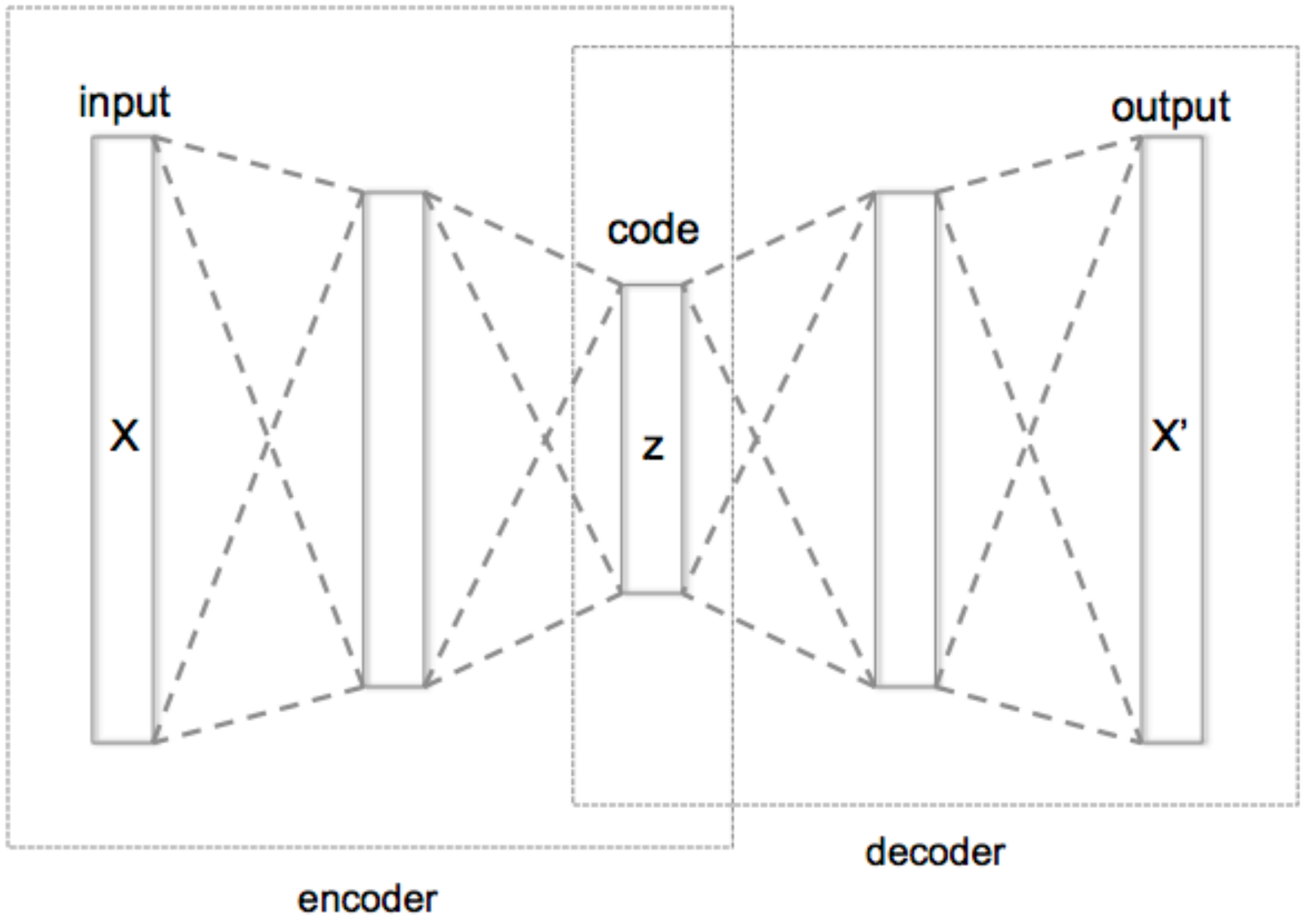


Time Series

Models

- The interpretability methods I talk about are **model dependent.** That means they use specific aspects of the insides of models to allow interpretability

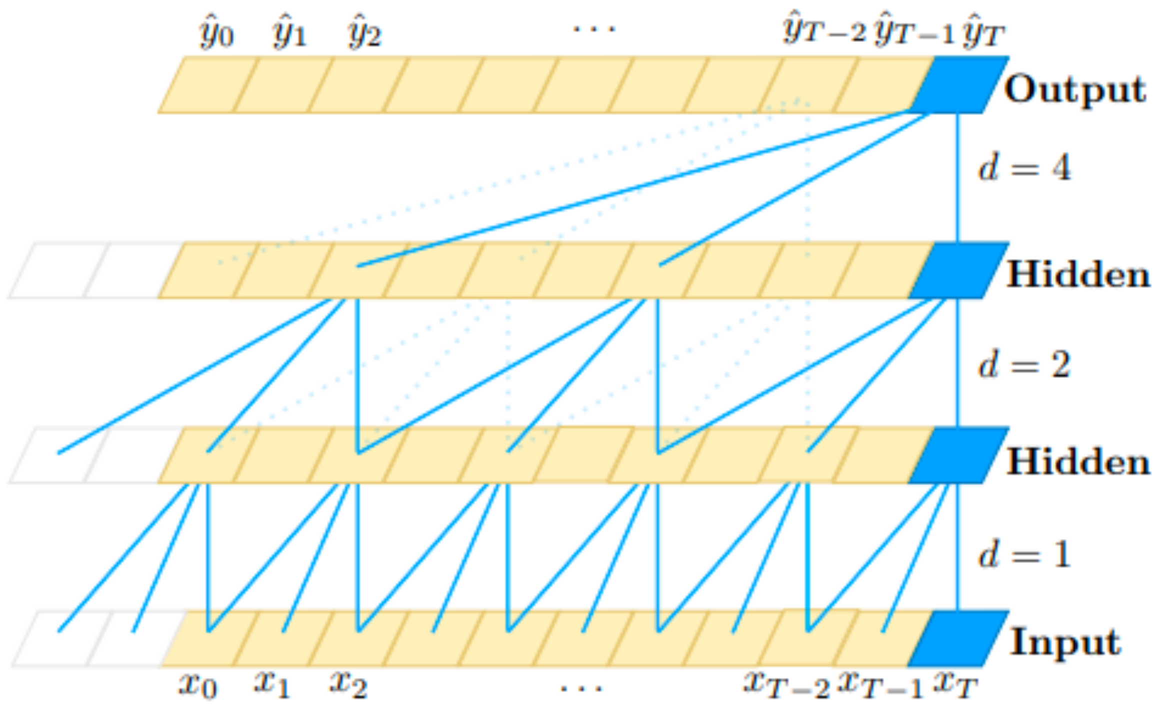
- Auto-Encoders



Time Series

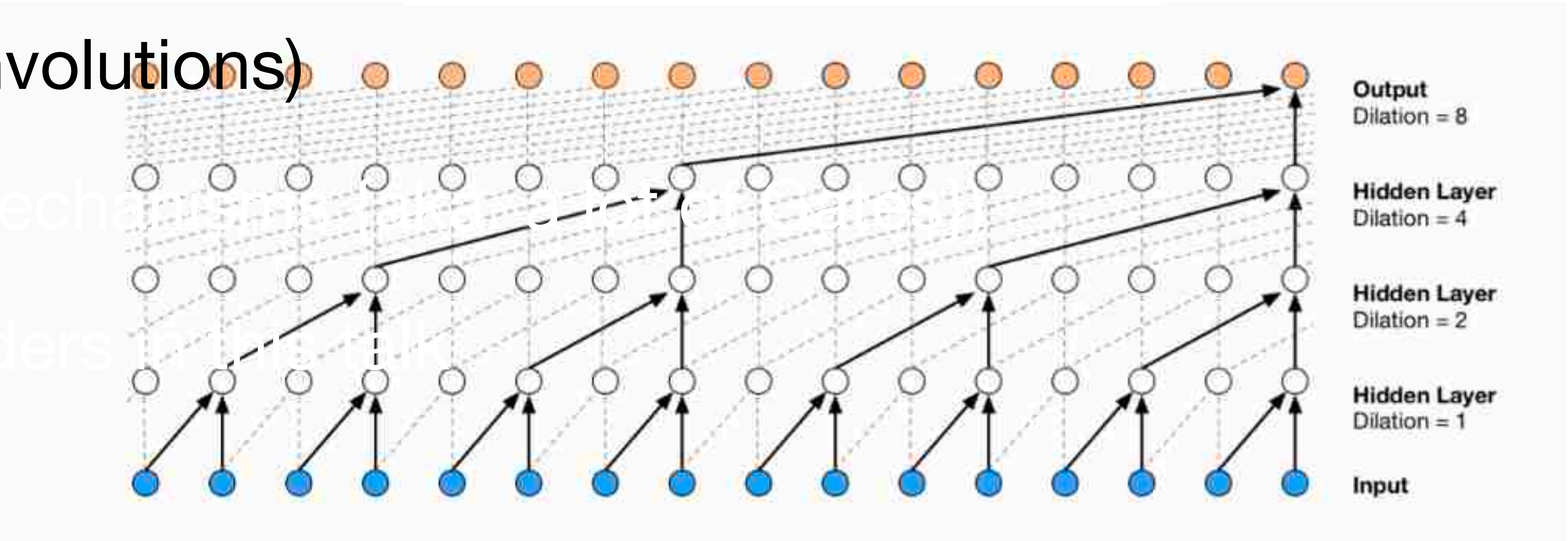
Models

- The interpretability methods I talk about are **model dependent**. That means they use specific aspects of the insides of models to allow interpretability



- Temporal Convolution Network (TCN),

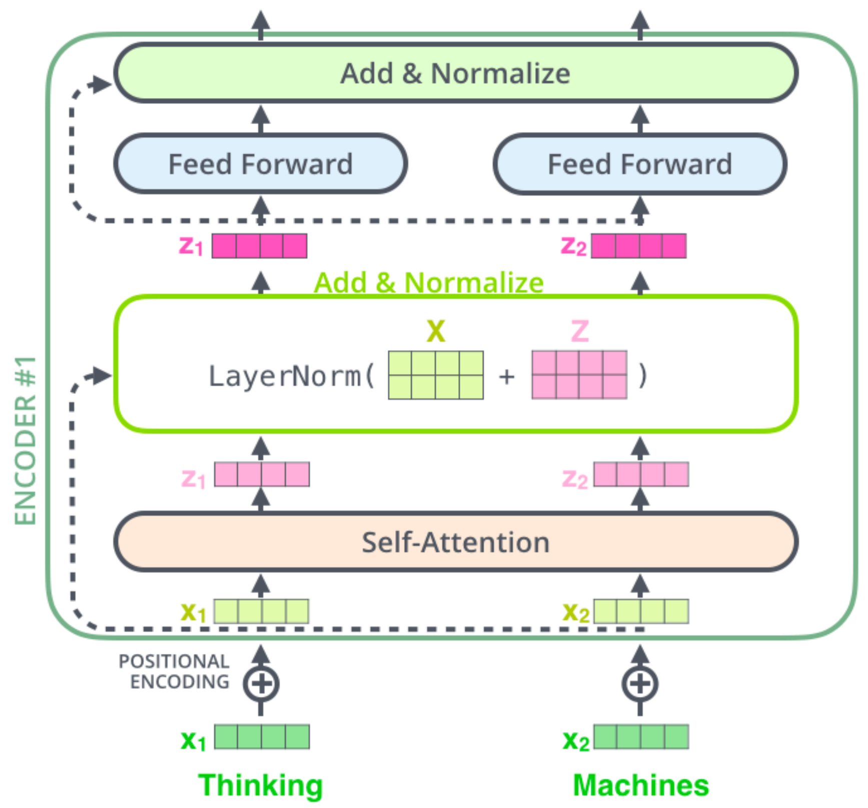
- WaveNet (Gated non-overlapping Convolutions)



Time Series

Models

- The interpretability methods I talk about are **model dependent**. That means they use specific aspects of the insides of models to allow interpretability



- Transformers (Multi-Head-Attention Mechanisms {aka. a lot of Gates})

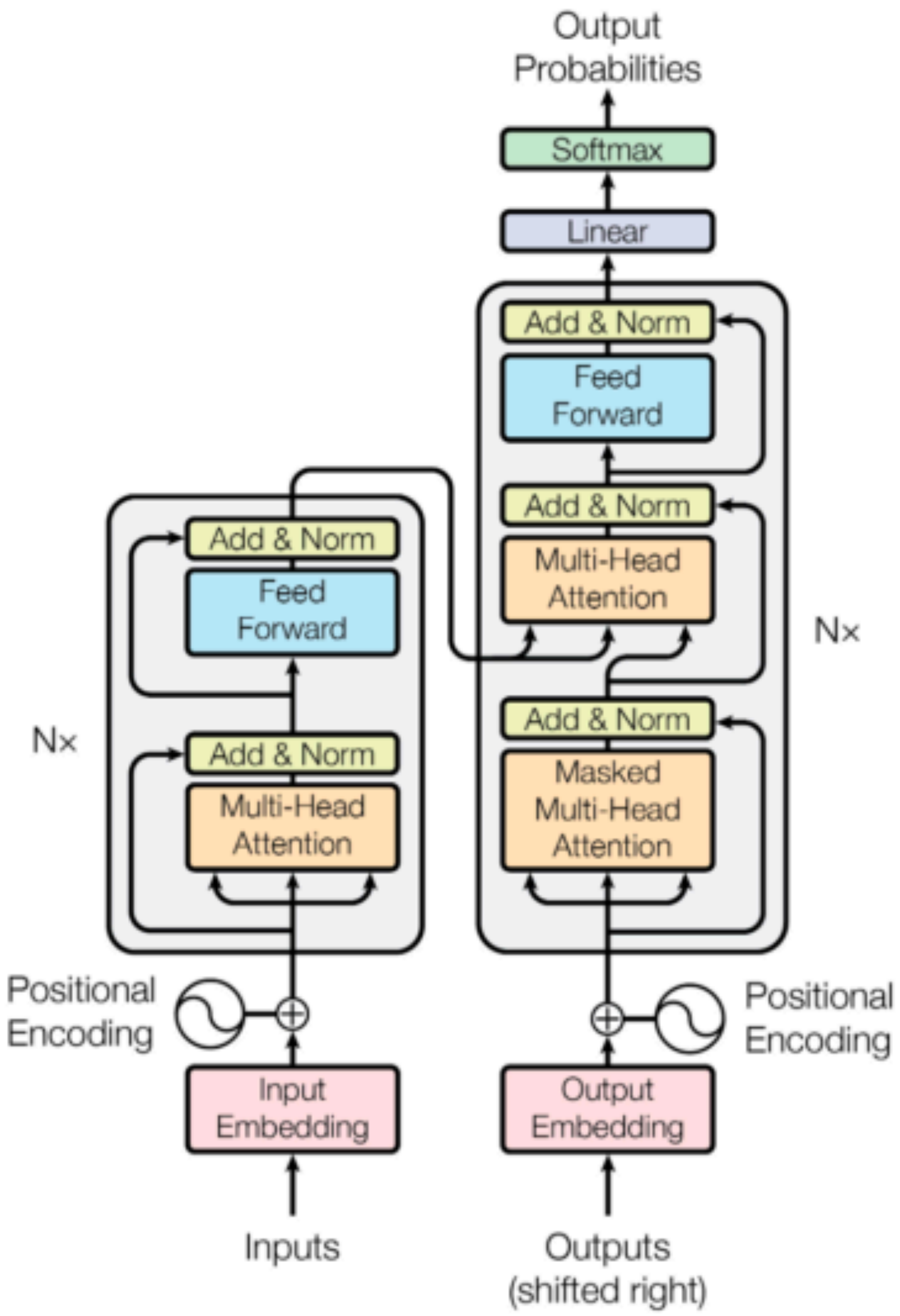


Figure 1: The Transformer - model architecture.

Time Series

Models

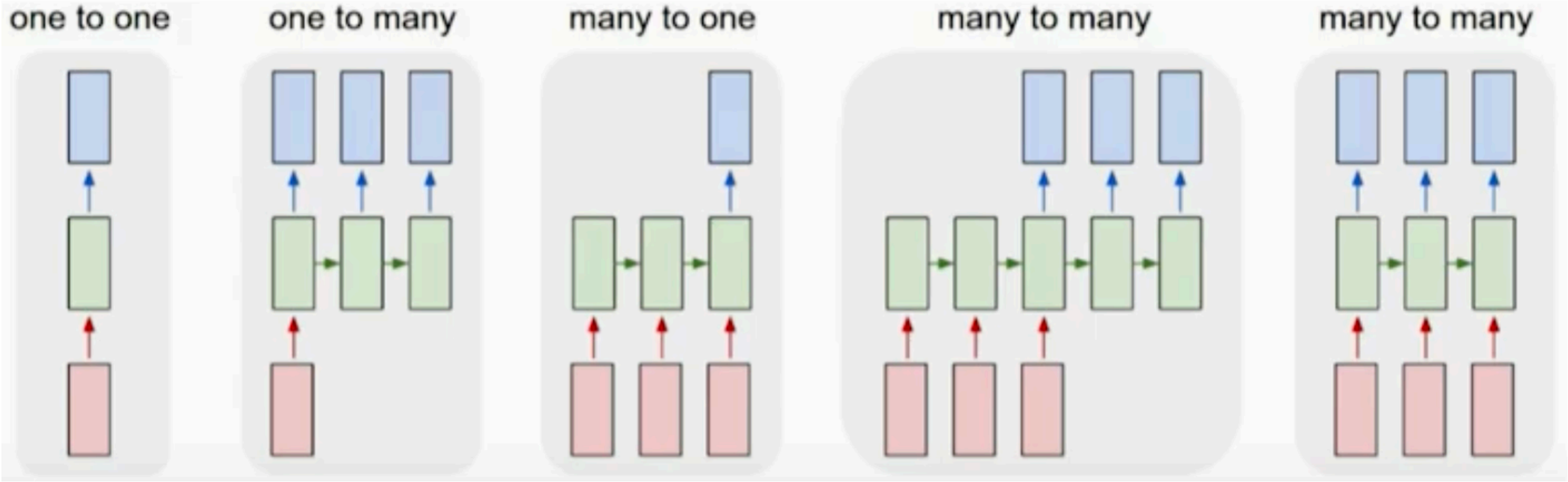
- The interpretability methods I talk about are **model dependent**. That means they use specific aspects of the insides of models to allow interpretability
- RNNs: **LSTMs**, GRUs, …
- Auto-Encoders
- Temporal Convolution Network (TCN), WaveNet (Gated non-overlapping Convolutions)
- Transformers (Multi-Head-Attention Mechanisms {aka. a lot of Gates})
- Hybrids: We also look at LSTM+Encoders in this talk

Time Series

Models

- The interpretability methods I talk about are **model dependent**. That means they use specific aspects of the insides of models to allow interpretability
- RNNs: **LSTMs**, GRUs, …
- Auto-Encoders
- Temporal Convolution Network (TCN), WaveNet (Gated non-overlapping Convolutions)
- Transformers (Multi-Head-Attention Mechanisms {aka. a lot of Gates})
- Hybrids: We also look at LSTM+Encoders in this talk

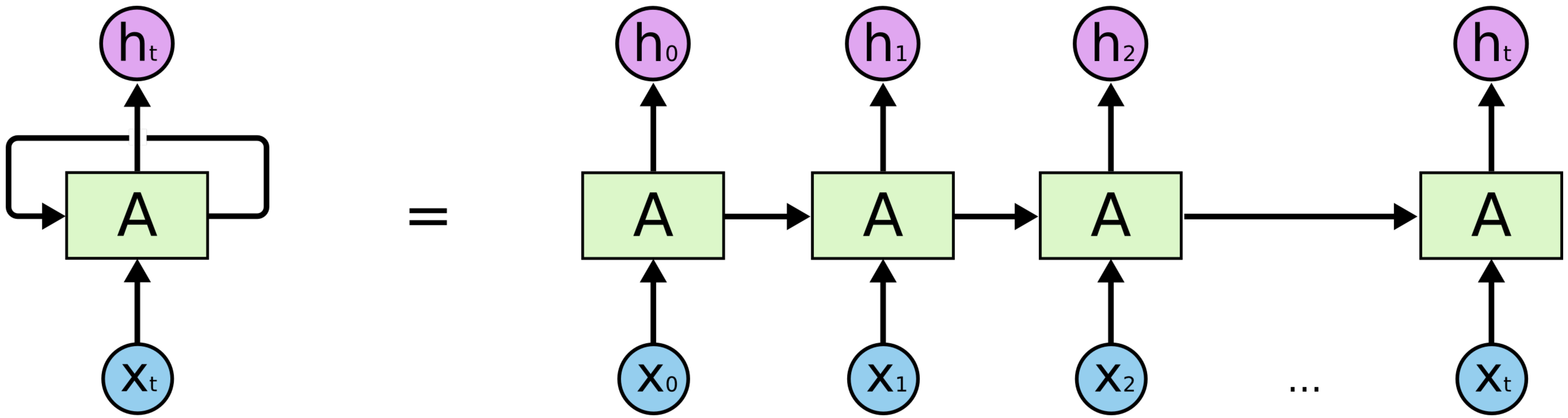
Time Series
Models



Recurrent Neural Networks (RNN)

Quick Recap

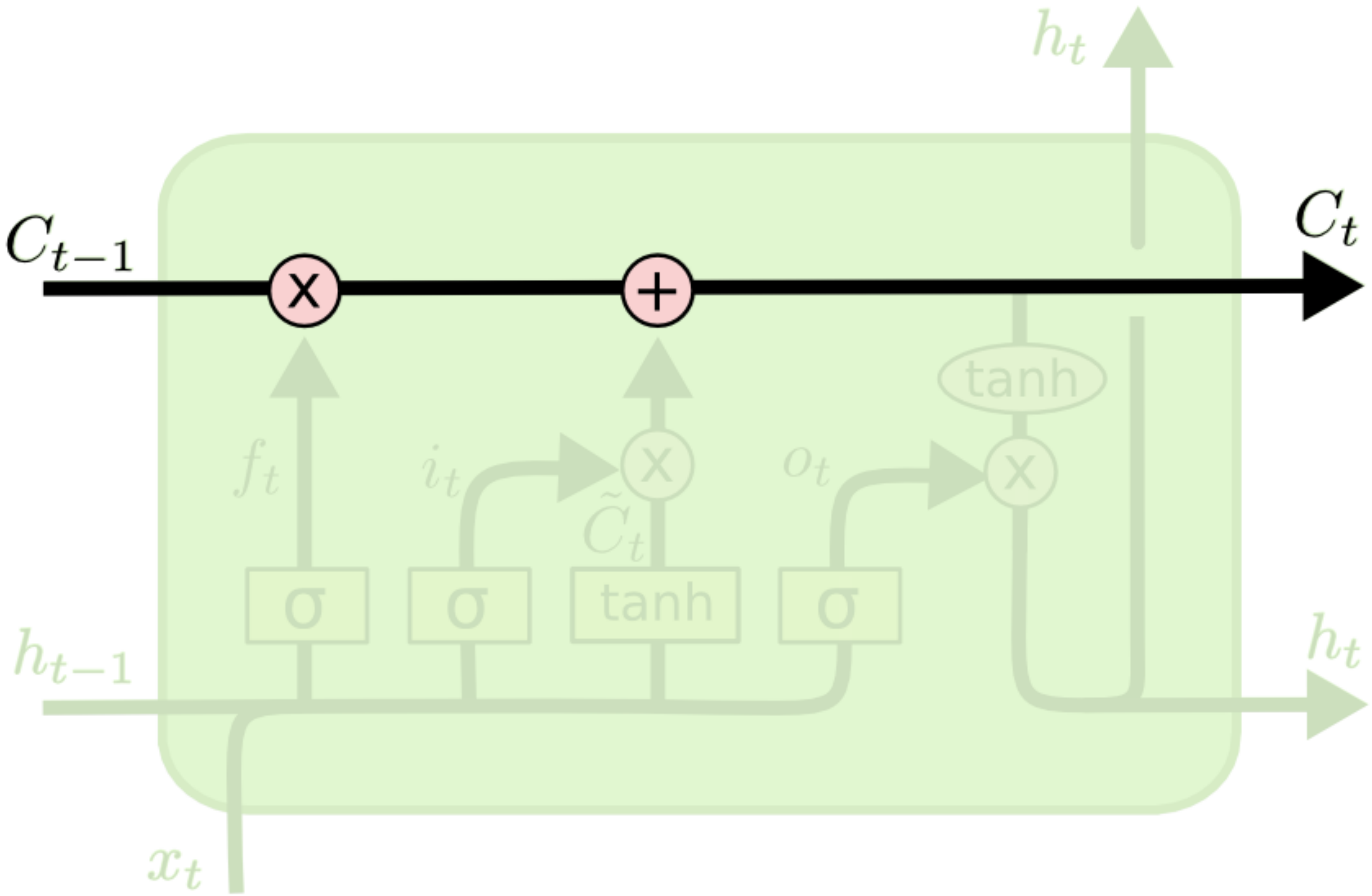
$$
\begin{aligned}
\boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \\
\boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}) \\
\boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \\
\hat{\boldsymbol{y}}^{(t)} &= \mathrm{softmax}(\boldsymbol{o}^{(t)})
\end{aligned}
\qquad (10.6)
$$



Long-Short Term Memory - LSTM

It's inner workings

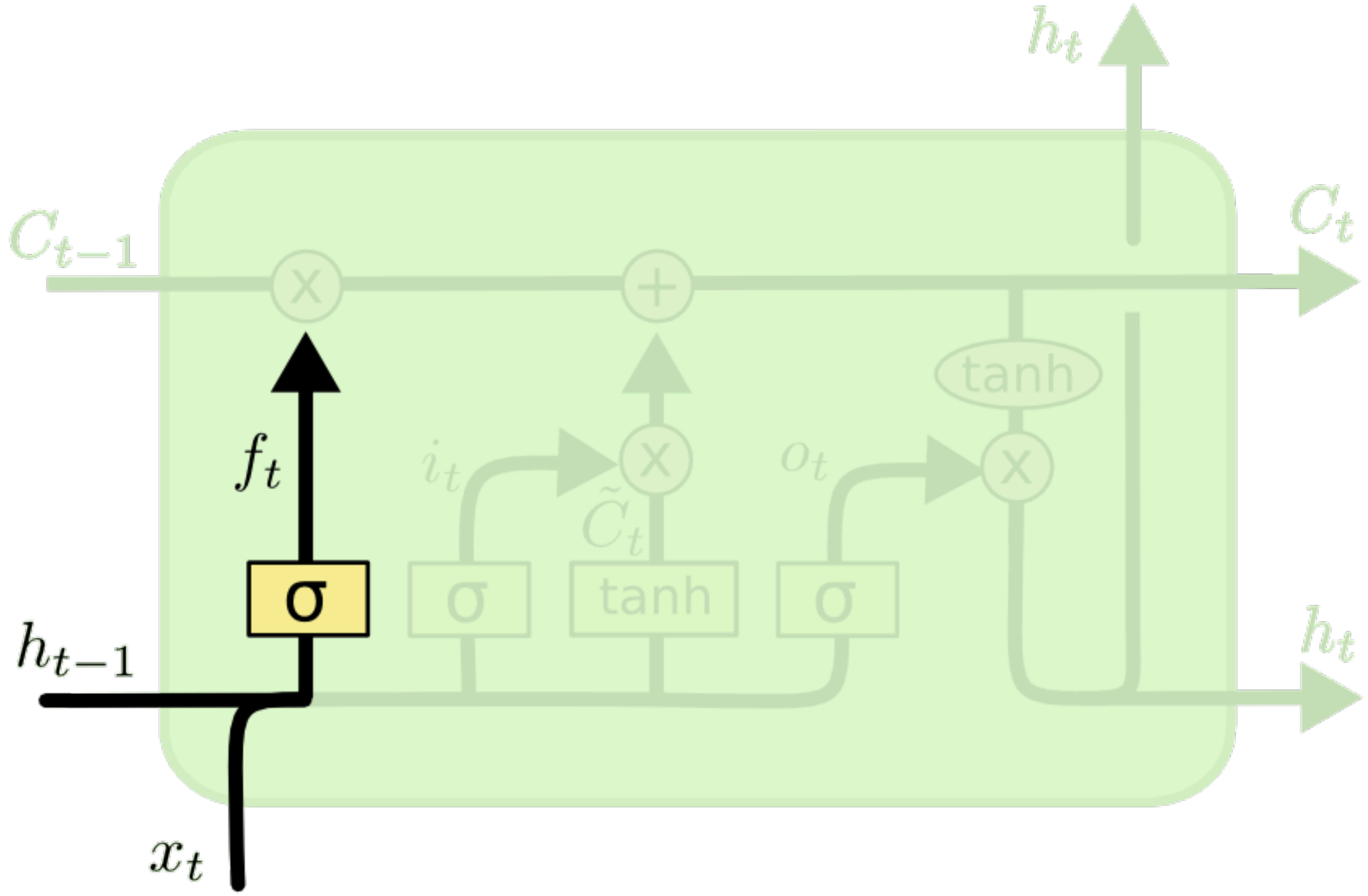
From: https://colah.github.io/posts/2015-08-Understanding-LSTMs/



Long-Short Term Memory - LSTM

It's inner workings

From: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

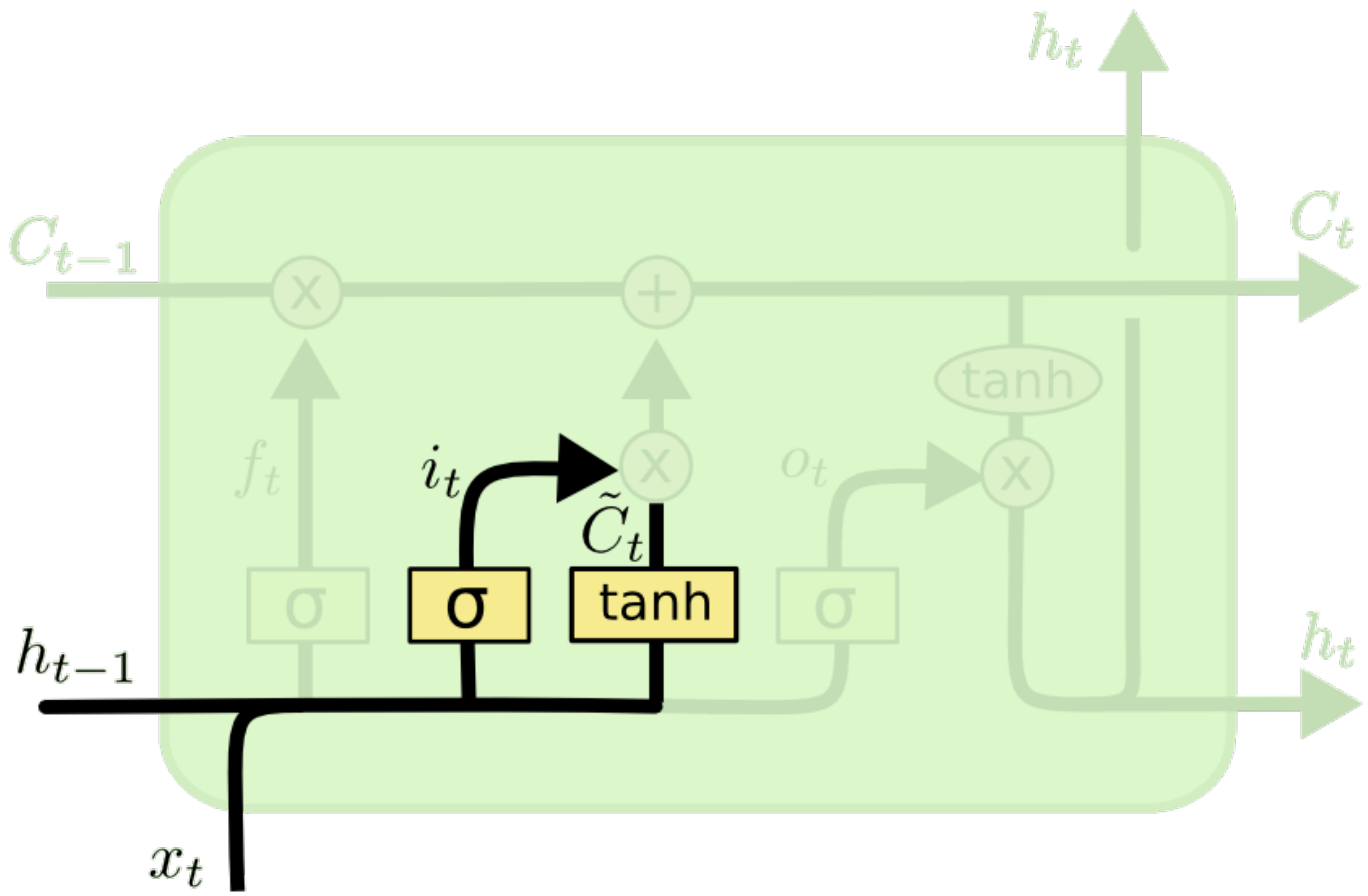


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Long-Short Term Memory - LSTM

It's inner workings

From: https://colah.github.io/posts/2015-08-Understanding-LSTMs/



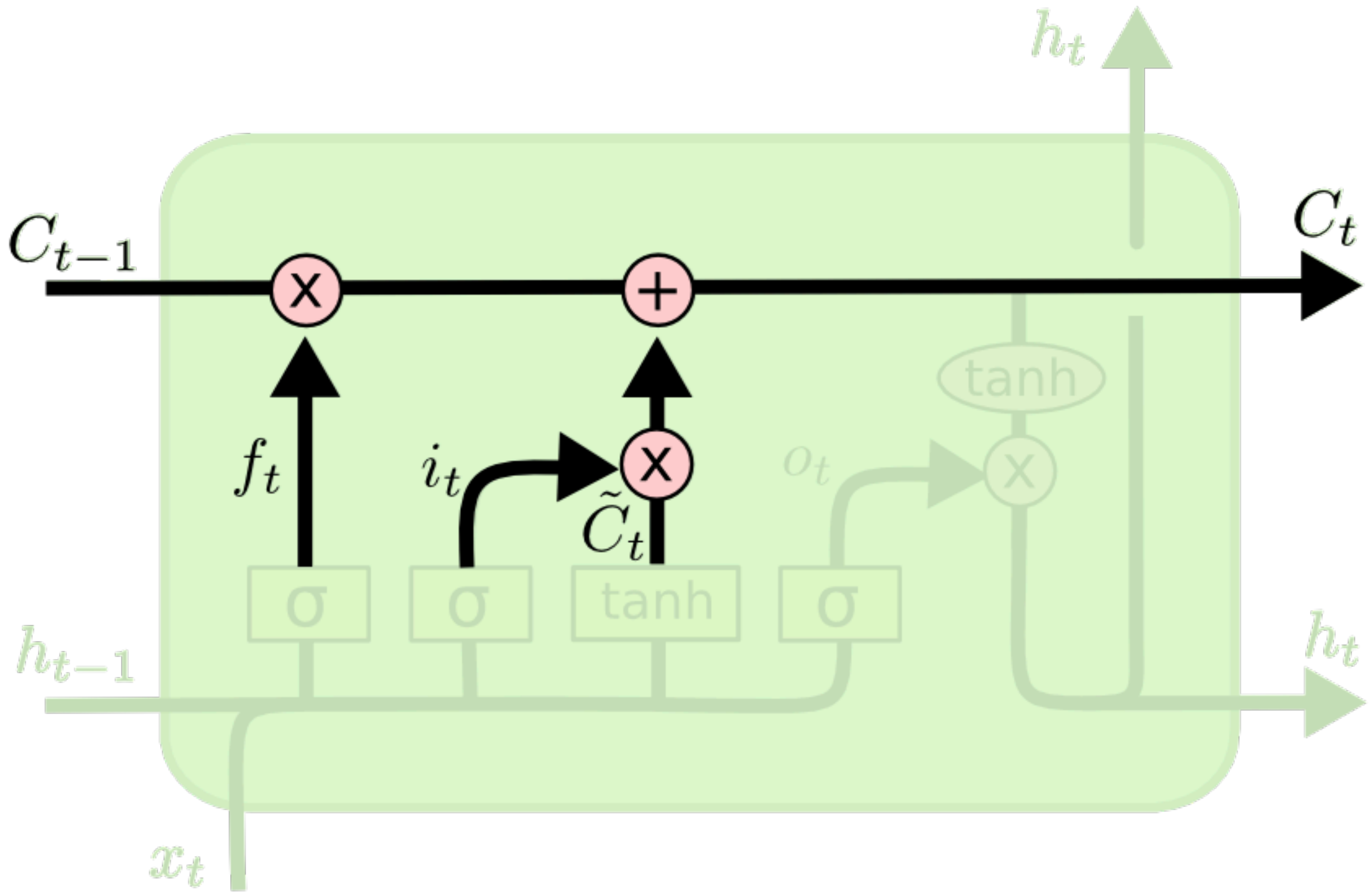
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Long-Short Term Memory - LSTM

It's inner workings

From: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

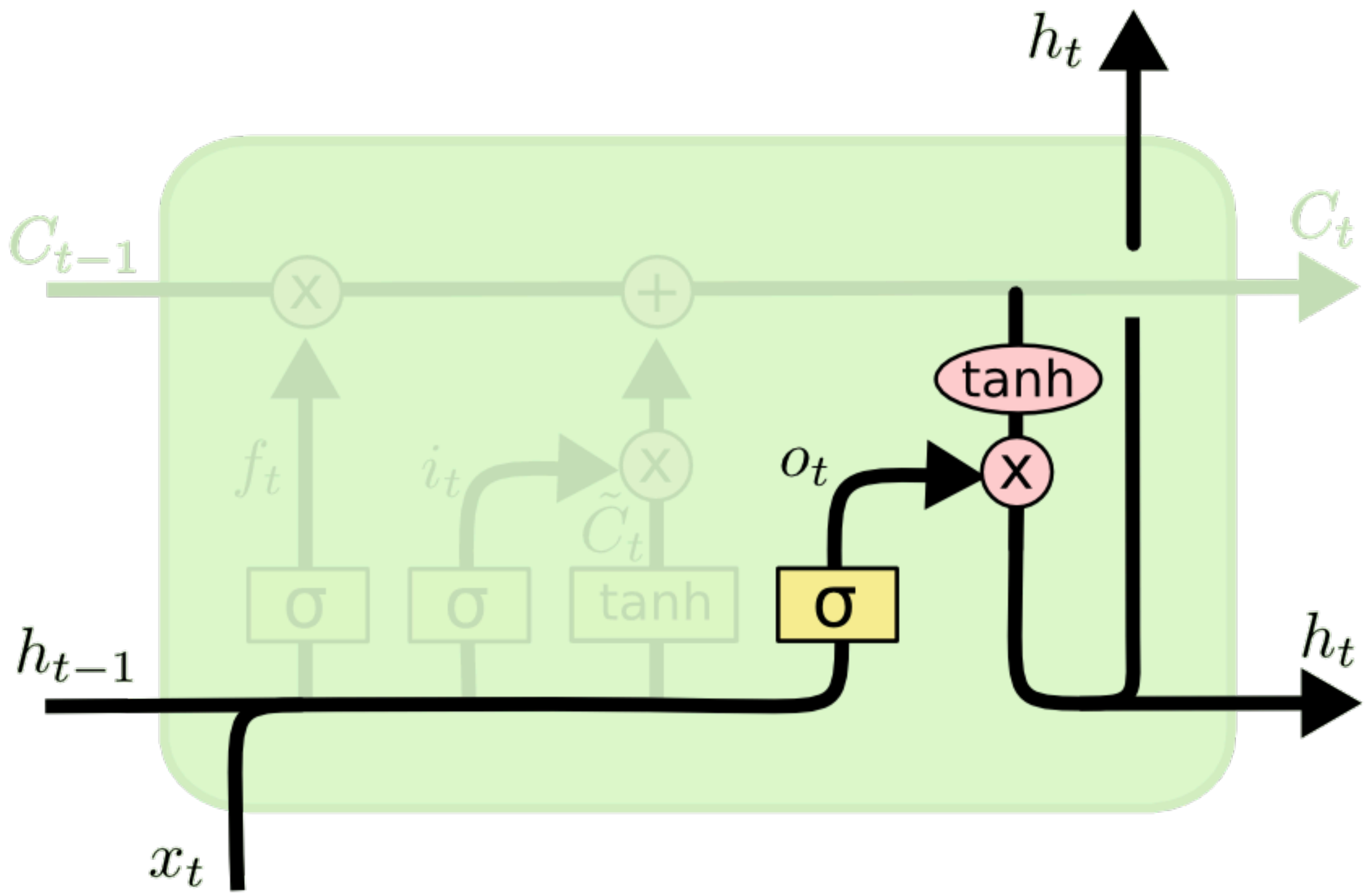


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Long-Short Term Memory - LSTM

It’s inner workings

From: https://colah.github.io/posts/2015-08-Understanding-LSTMs/



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Demo Time

https://github.com/grazai/xai-tutorial-april-2020

Outlook

Probable next Talks

- LRP by Ilija
- Finish LSTM:
  - Embeddings
  - LRP for LSTM
- Convolutions and Time (TCN, WaveNet)
- Transformers & Attention Mechanisms
- DeepFeature
- ReMIX

Thanks for the Attention

We will continue for an Q&A in discord (we will post the link in the twitch chat)