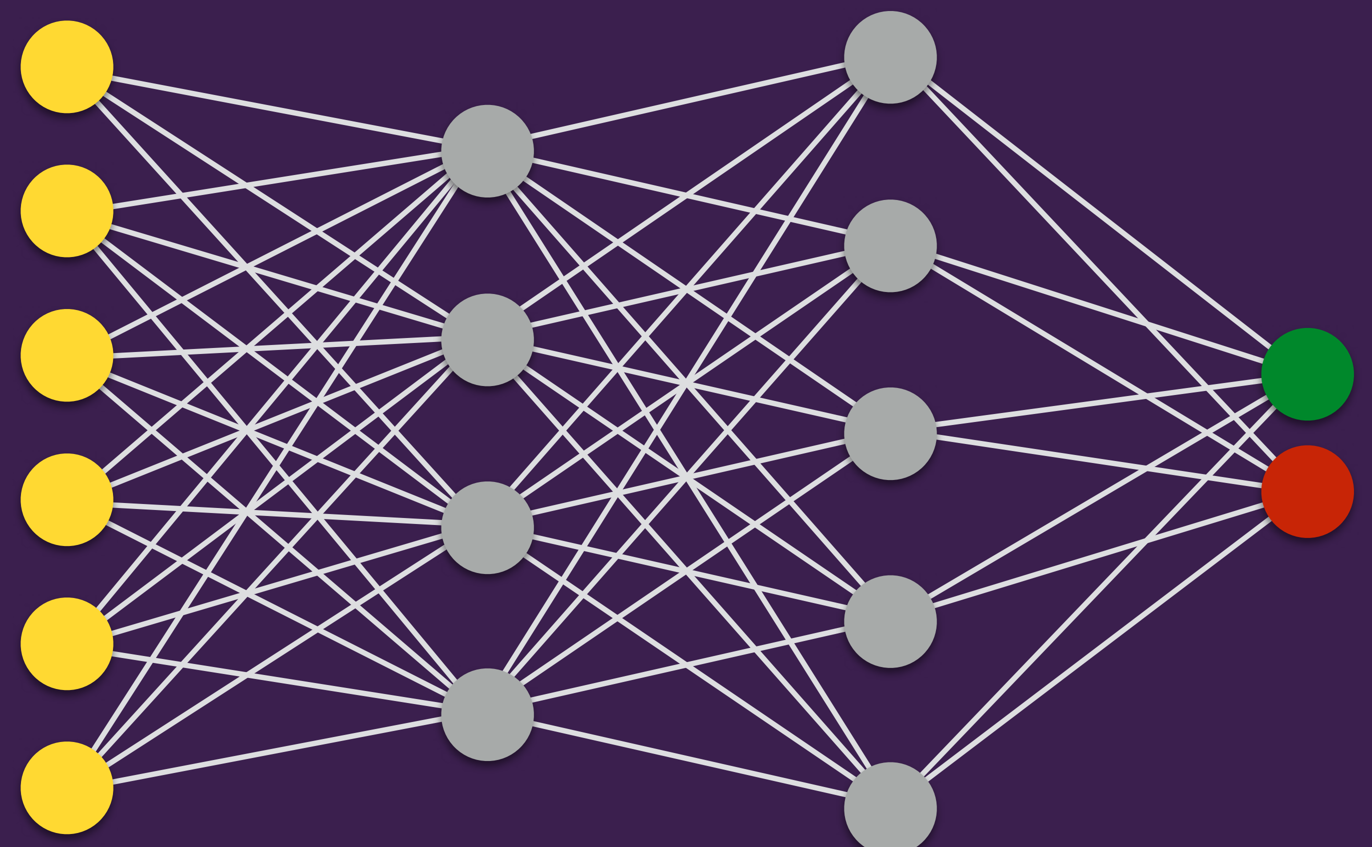
Perfectly Parallel Fairness Certification of Neural Networks



Caterina Urban, Maria Christakis, Valentin Wüstholz, Fuyuan Zhang


WIRED
BUSINESS
MORE
SIGN IN
SUBSCRIBE
ERIC NILER
BUSINESS
03.25.2019 07:00 AM
Can AI Be a Fair Judge in Court? Estonia Thinks So
Estonia plans to use an artificial intelligence program to decide some small-claims cases, part of a push to make government services smarter.
Machine Bias
There's software used across the country to predict future criminals. And it's biased against blacks.
by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica
May 23, 2016
WIRED
In 2019, predictive algorithms will start to make banking fair for all
AUTOMATED BACKGROUND CHECKS ARE DECIDING WHO'S FIT FOR A HOME
By Colin Lecher | @colinlecher |
China 'social credit': Beijing sets up huge system
By Celia Hatton
BBC News, Beijing
26 October 2015
4 ways to check for skin cancer with your smartphone
Your phone can help you recognize suspicious moles and marks, but you should still see a dermatologist or doctor.
BY AMANDA CAPRITTO | SEPTEMBER 16, 2019 10:57 AM PDT
STAT+
IBM's Watson supercomputer recommended 'unsafe and incorrect' cancer treatments, internal documents show
By Casey Ross @caseymross and Ike Swetlitz
July 25, 2018
Google Translate
Text
Documents
DETECT LANGUAGE
ENGLISH
FRENCH
ENGLISH
A nurse
A doctor
Une infirmière
Un médecin
16/5000
WIRED
TOM SIMONITE
BUSINESS
12.21.2019 08:00 AM
The AI Doctor Will See You Now
nature
Search
NEWS · 24 OCTOBER 2019
UPDATE 26 OCTOBER 2019
Millions of black people affected by racial bias in health-care algorithms
Study reveals rampant racism in decision-making software used by US hospitals – and highlights ways to correct it.
BUSINESS NEWS
OCTOBER 10, 2018 / 5:12 AM / A YEAR AGO
Amazon scraps secret AI recruiting tool that showed bias against women
Jeffrey Dastin
AI used for first time in job interviews in UK to find best applicants
By Charles Hymas
27 SEPTEMBER 2019 • 10:00 PM

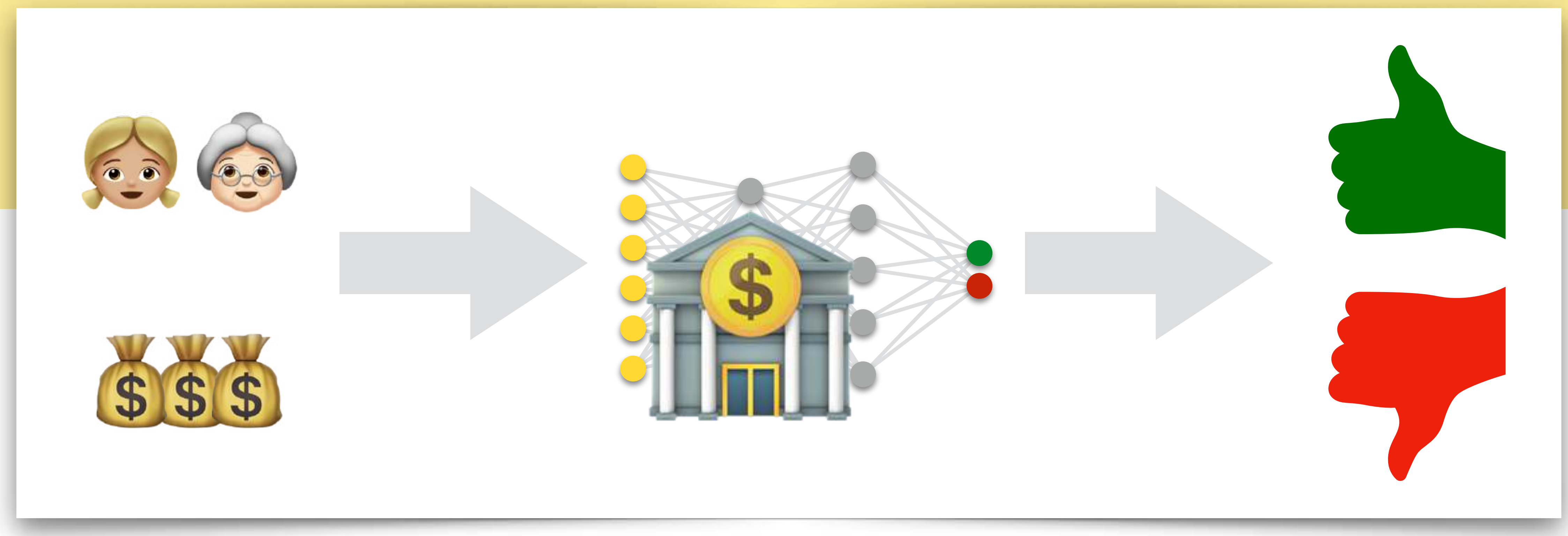
Fairness Certification of Machine Learning Systems is Now Critical!

Feed-Forward Neural Networks

Classification of Tabular Data

increasingly used for:
- very large datasets (i.e., billions of rows)
- data that comes in batches over time

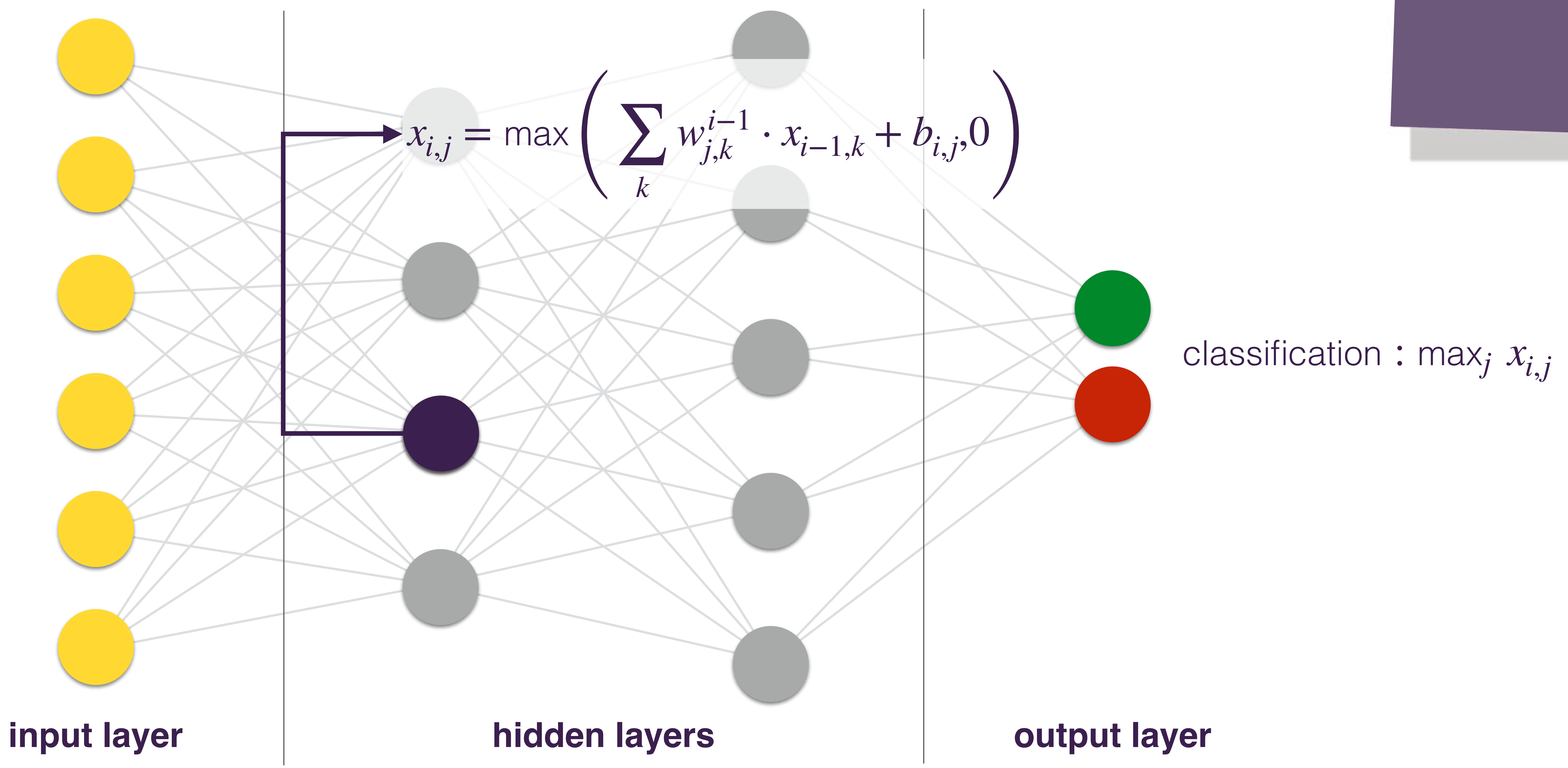
Fairness Certification of Machine Learning Systems is Now Critical!



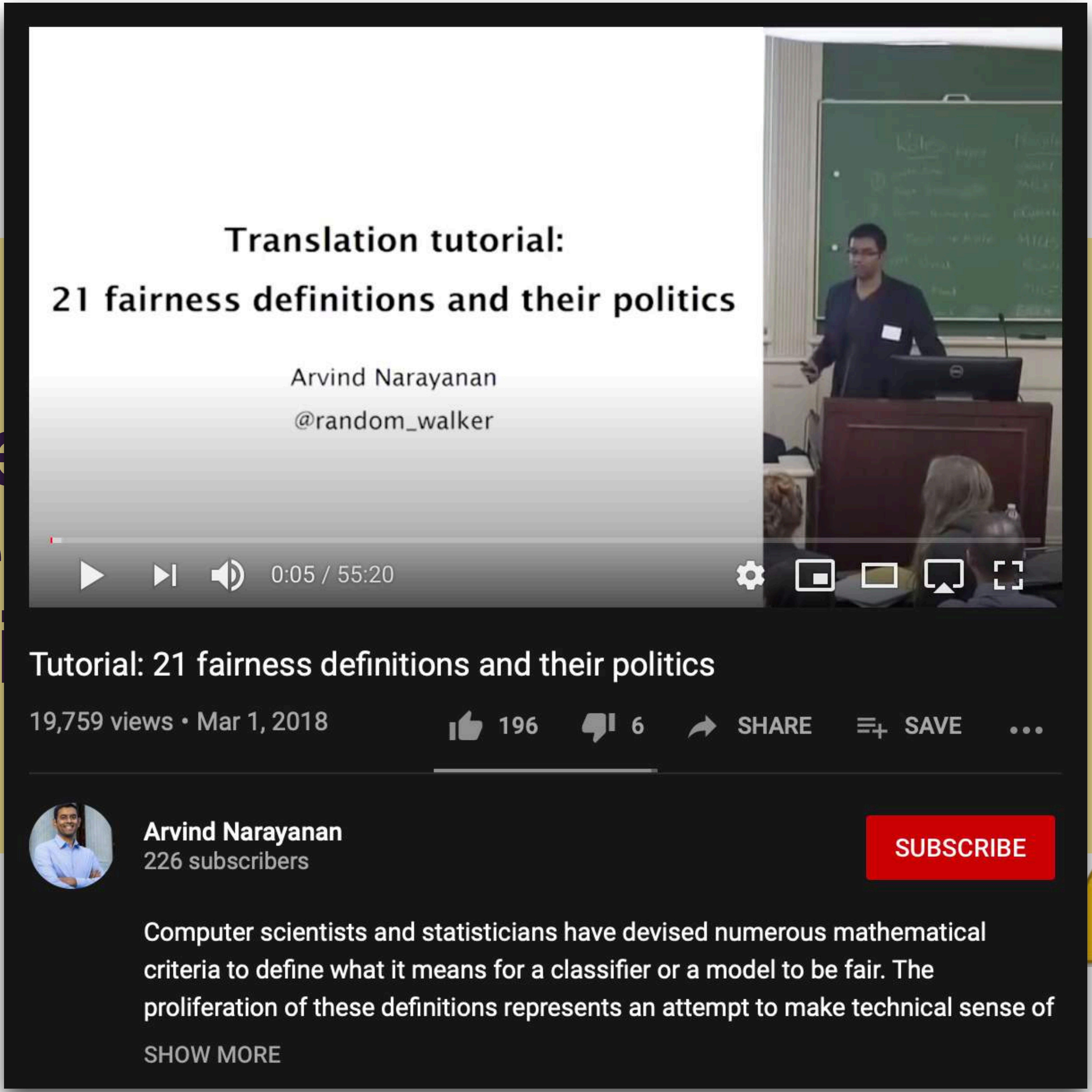
Feed-Forward Neural Networks

with ReLU Activations

other activation functions are discussed in the paper



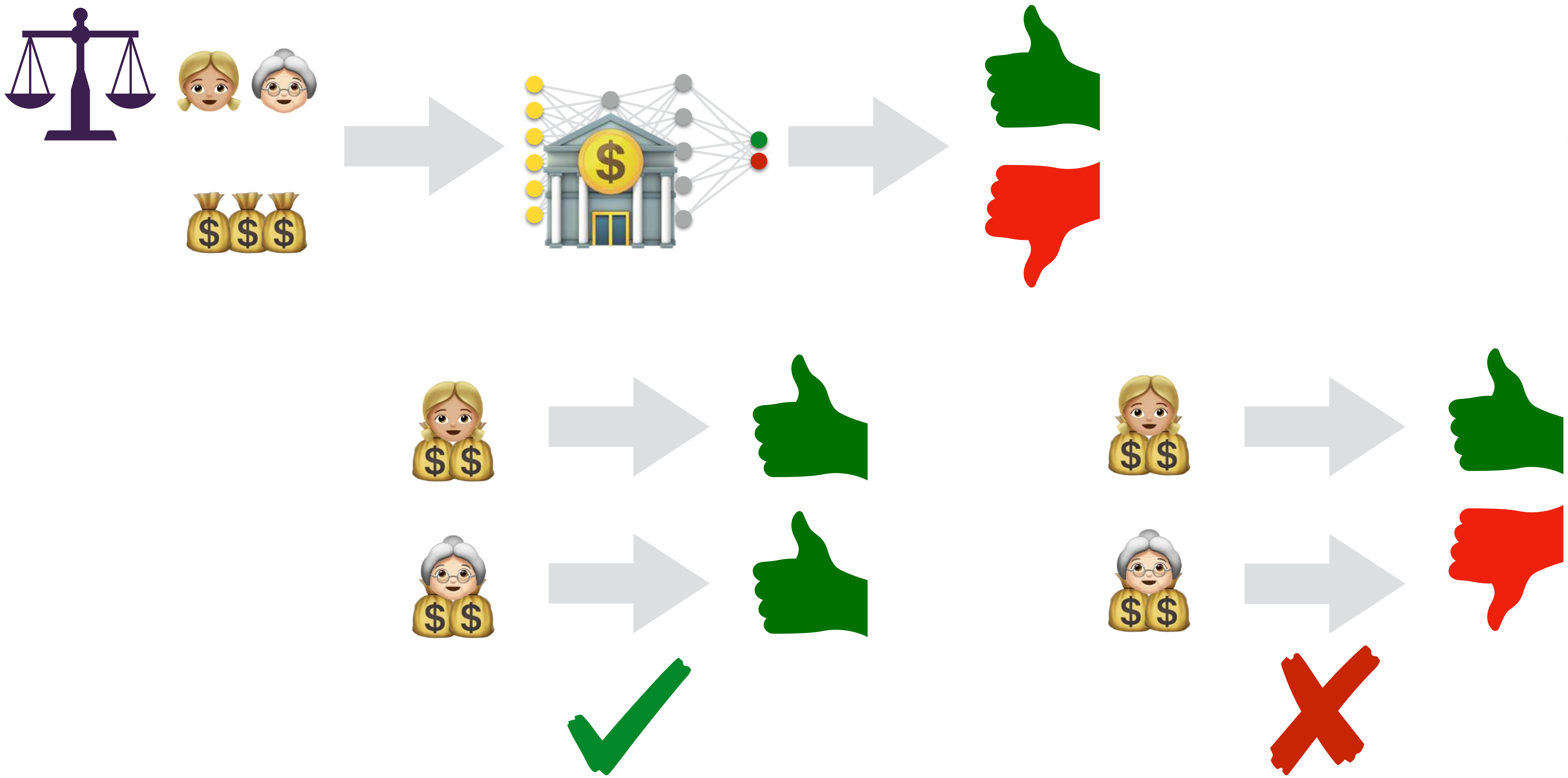
Fairness Ce
of Machine
is Now Criti



Dependency Fairness

the output classification is independent of the values of the sensitive input feature(s)

- does not require an **oracle**
- amenable to **static analysis**
- stronger than **group fairness**



Galhotra et al. - Fairness Testing: Testing Software for Discrimination (FSE 2017)

Static Analysis by Abstract Interpretation

AIRBUS

AREVA

HELBAKO

Fairness Certification of Machine Learning Systems

ABSTRACT INTERPRETATION : A UNIFIED LATTICE MODEL FOR STATIC ANALYSIS OF PROGRAMS BY CONSTRUCTION OR APPROXIMATION OF FIXPOINTS

Patrick Cousot* and Radhia Cousot**

Laboratoire d'Informatique, U.S.M.G., BP. 53
38041 Grenoble cedex, France

1. Introduction

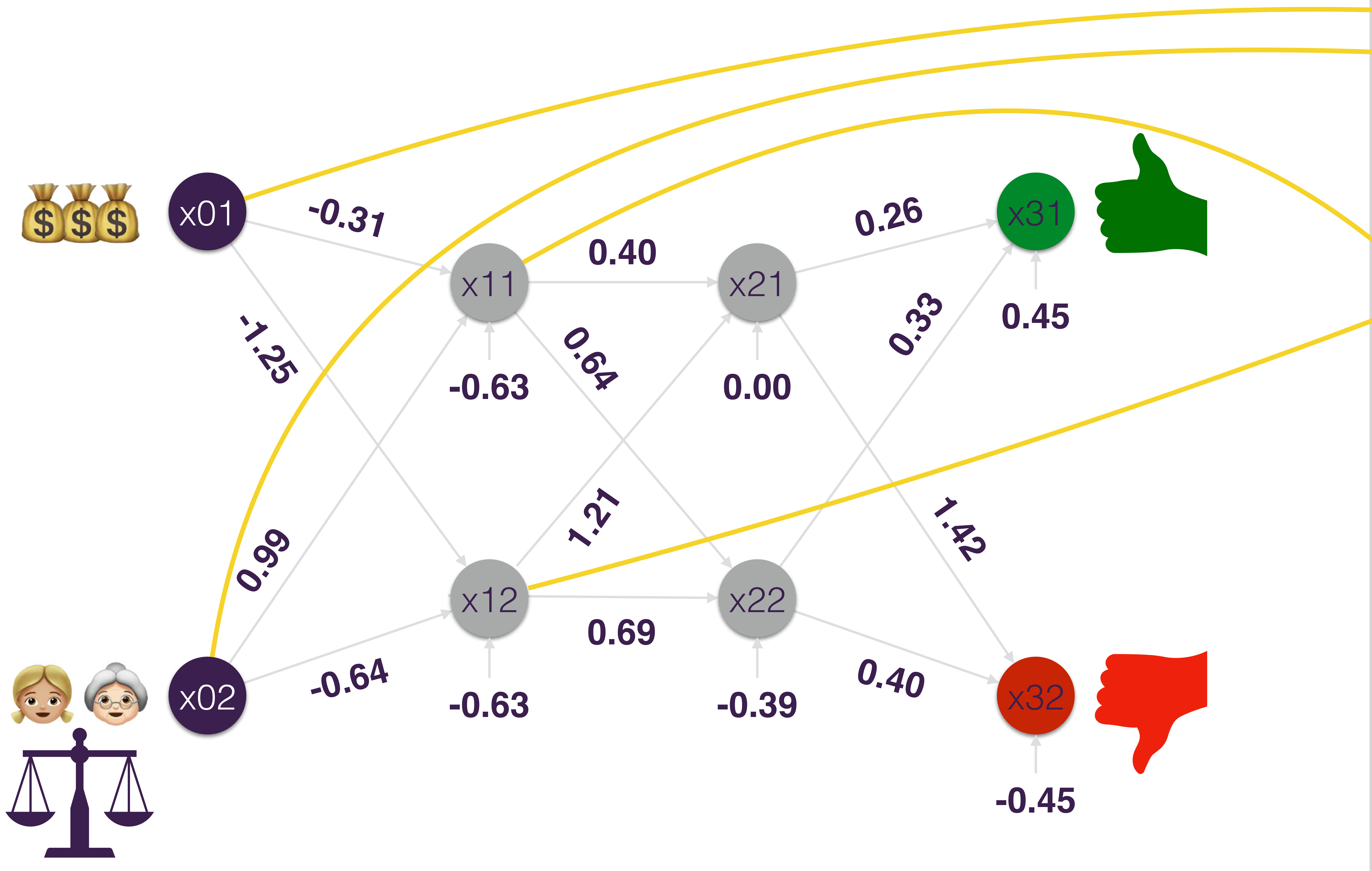
A program denotes computations in some universe of objects. Abstract interpretation of programs consists in using that denotation to describe computations in another universe of abstract objects, so that the results of abstract execution give some informations on the actual computations. An

Abstract program properties are modeled by a complete semilattice, Birkhoff[61]. Elementary program constructs are locally interpreted by order preserving functions which are used to associate a system of recursive equations with a program. The program global properties are then defined as one of the extreme fixpoints of that system, Tarski[55]. The abstraction process is defined in section 6. It

Radhia Cousot

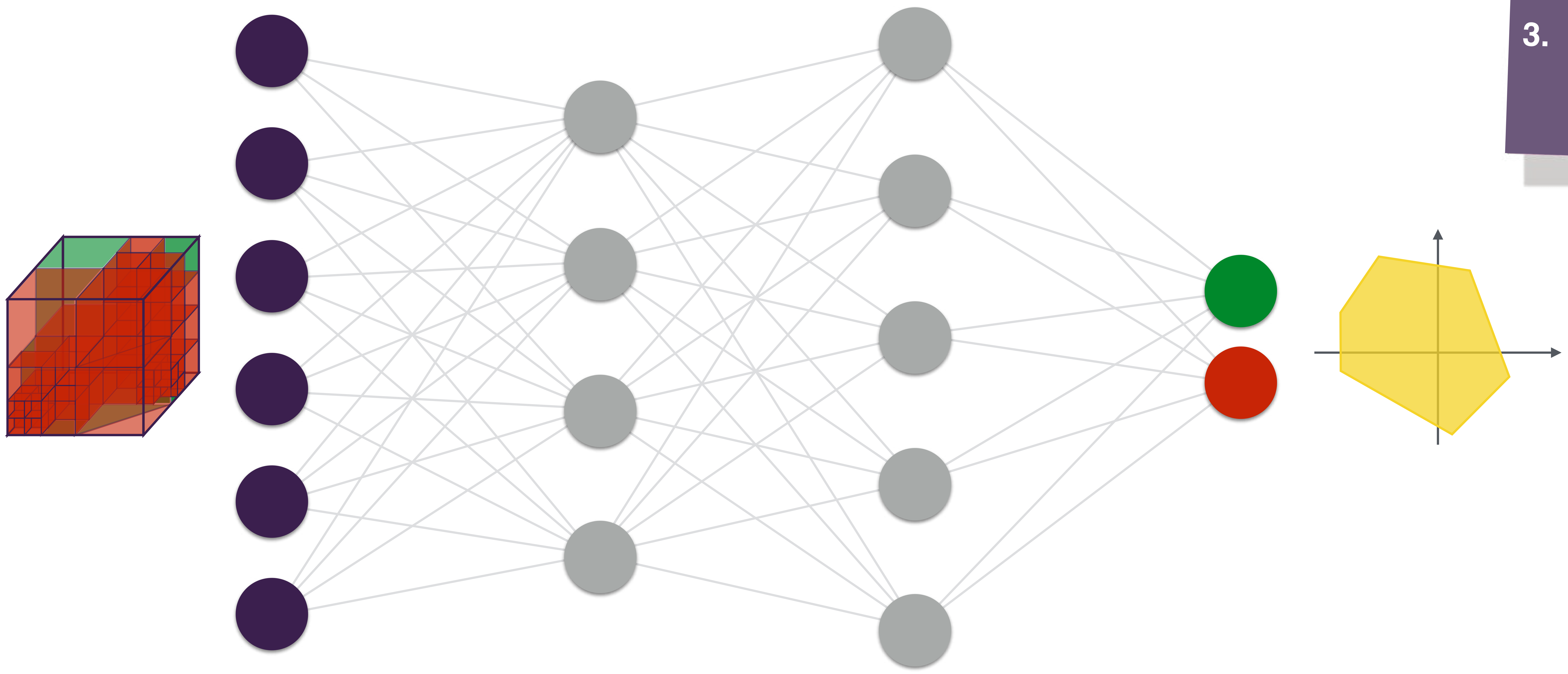
Patrick Cousot

Toy Example




```
x01 = input()
x02 = input()

x11 = -0.31 * x01 + 0.99 * x02 + (-0.63)
x12 = -1.25 * x01 + (-0.64) * x02 + 1.88

x11 = 0 if x11 < 0 else x11
x12 = 0 if x12 < 0 else x12

x21 = 0.40 * x11 + 1.21 * x12 + 0.00
x22 = 0.64 * x11 + 0.69 * x12 + (-0.39)

x21 = 0 if x21 < 0 else x21
x22 = 0 if x22 < 0 else x22

x31 = 0.26 * x21 + 0.33 * x22 + 0.45
x32 = 1.42 * x21 + 0.40 * x22 + (-0.45)

if x31 > x32:
   print('credit approved')
elif x32 < x31:
   print('credit denied')
```

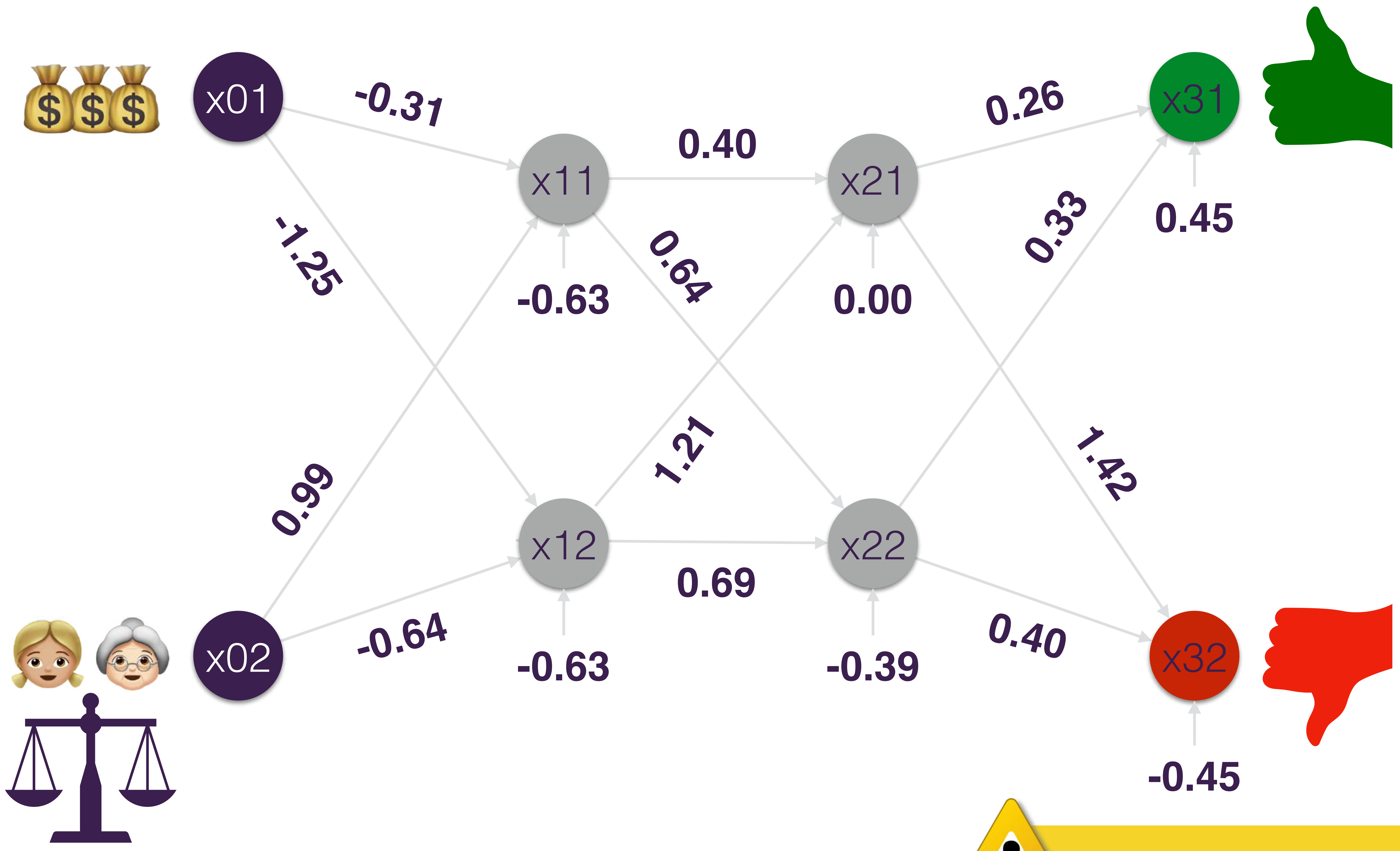
Naïve Backward Analysis



1. proceed **backwards** from all possible classifications
2. **project** away the value of the sensitive feature(s)
3. check for **intersection**:
   empty → ✓ fair
   otherwise → 🚨 alarm

Toy Example

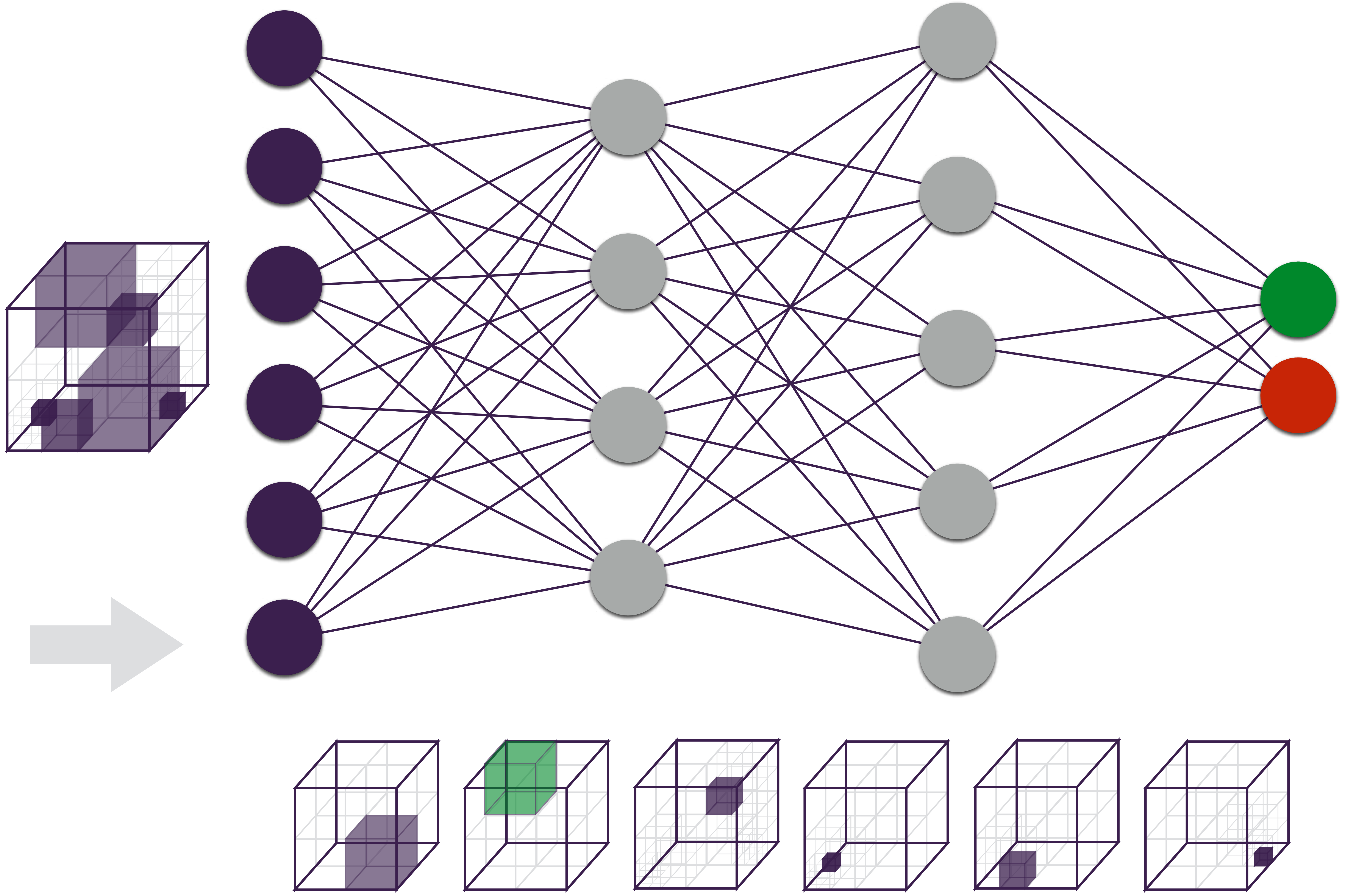
Naïve Backward Analysis



too many disjunctions!

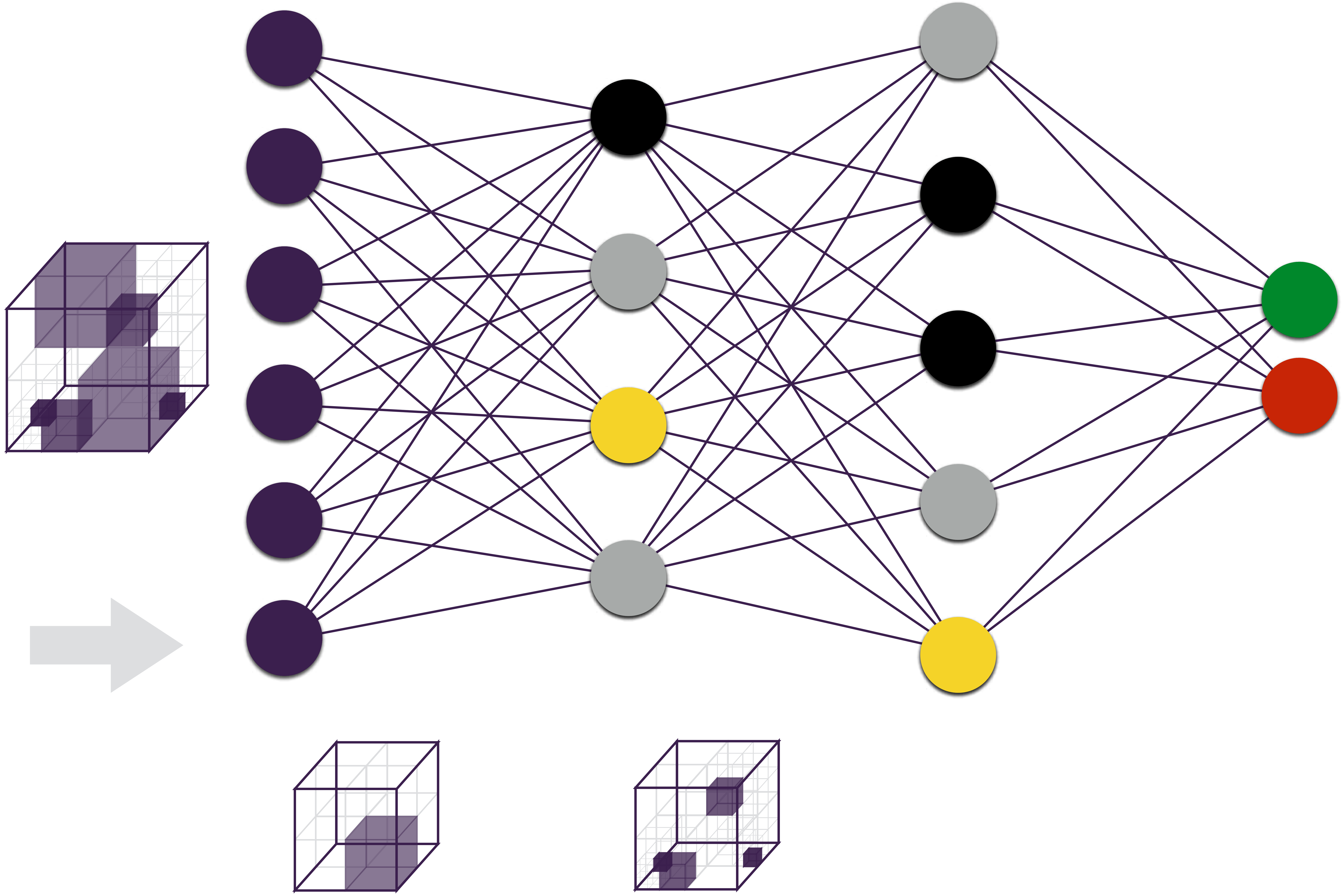
```
x01 = input()
x02 = input()

x11 = -0.31 * x01 + 0.99 * x02 + (-0.63)
x12 = -1.25 * x01 + (-0.64) * x02 + 1.88

x11 = 0 if x11 < 0 else x11
x12 = 0 if x12 < 0 else x12

x21 = 0.40 * x11 + 1.21 * x12 + 0.00
x22 = 0.64 * x11 + 0.69 * x12 + (-0.39)

x21 = 0 if x21 < 0 else x21
x22 = 0 if x22 < 0 else x22
1.16 * x21 + 0.07 * x22 < 0.90    1.16 * x21 + 0.07 * x22 > 0.90
x31 = 0.26 * x21 + 0.33 * x22 + 0.45
x32 = 1.42 * x21 + 0.40 * x22 + (-0.45)
x31 > x32    x32 > x31
if x31 > x32:
   print('credit approved')
elif x32 < x31:
   print('credit denied')
```

Our Solution



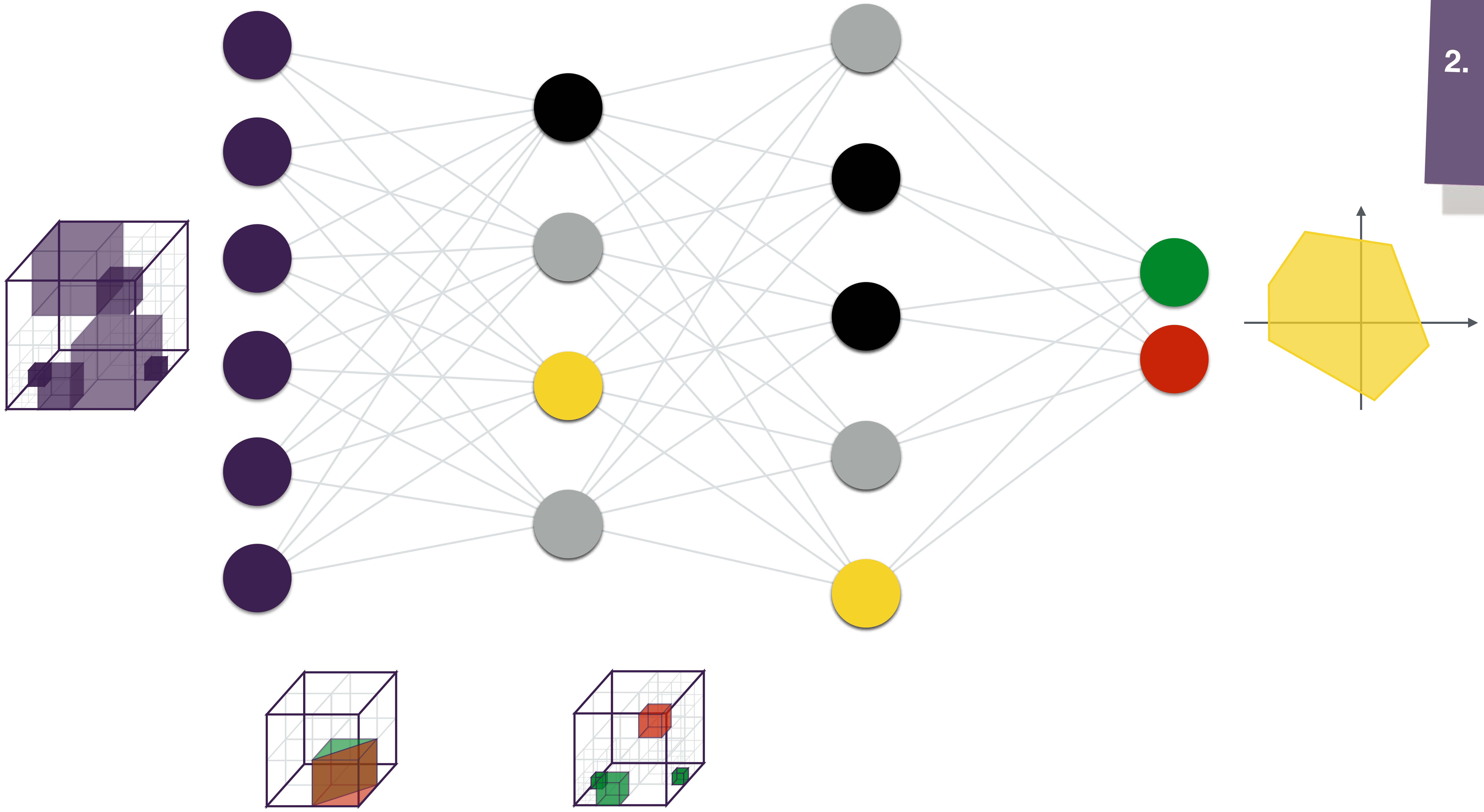
1. proceed **forwards** to find:
   - already ✓ **fair** partitions

Our Solution



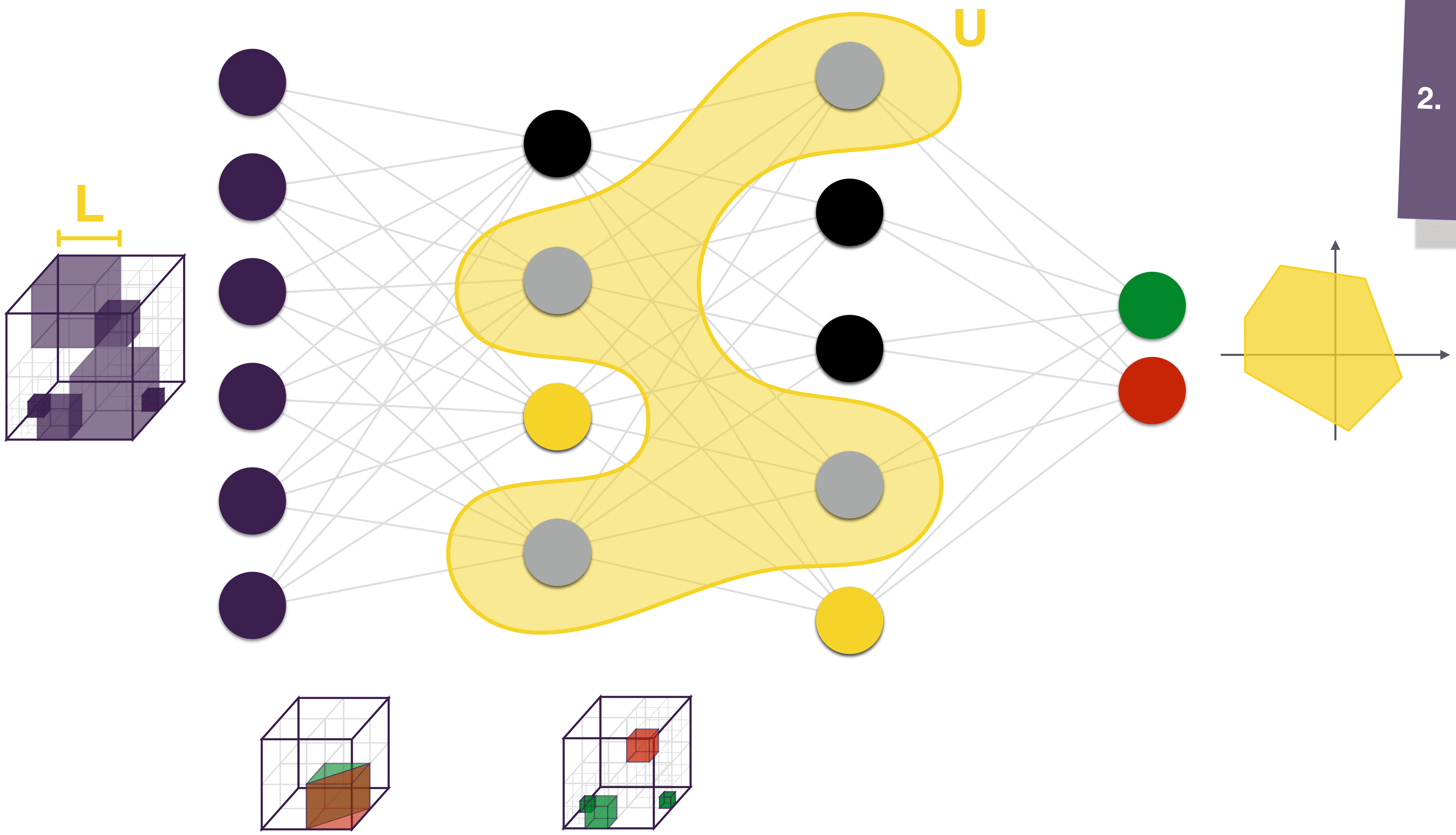
1. proceed forwards to find:
   - already ✓ fair partitions
   - activation patterns

Our Solution



1. proceed **forwards** to find:
   - already ✓ **fair** partitions
   - activation patterns
2. proceed **backwards** for each activation pattern

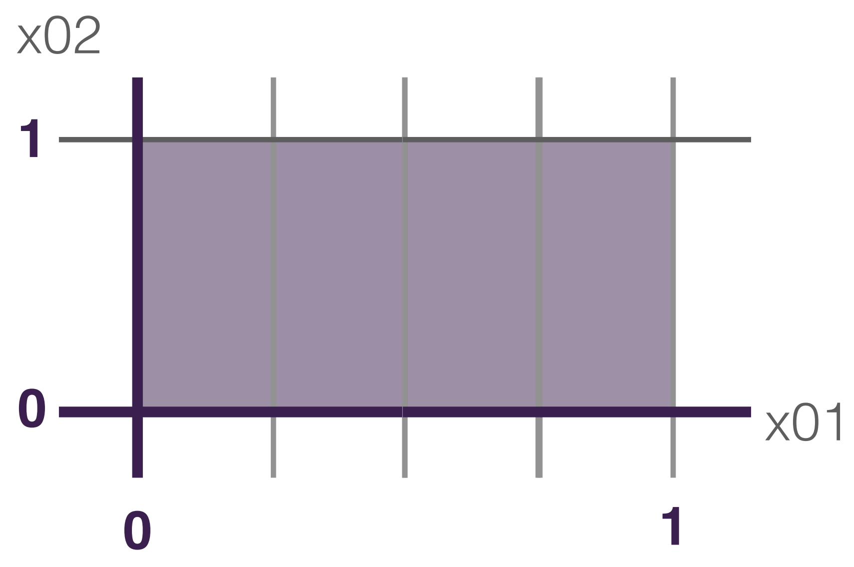
Our Solution

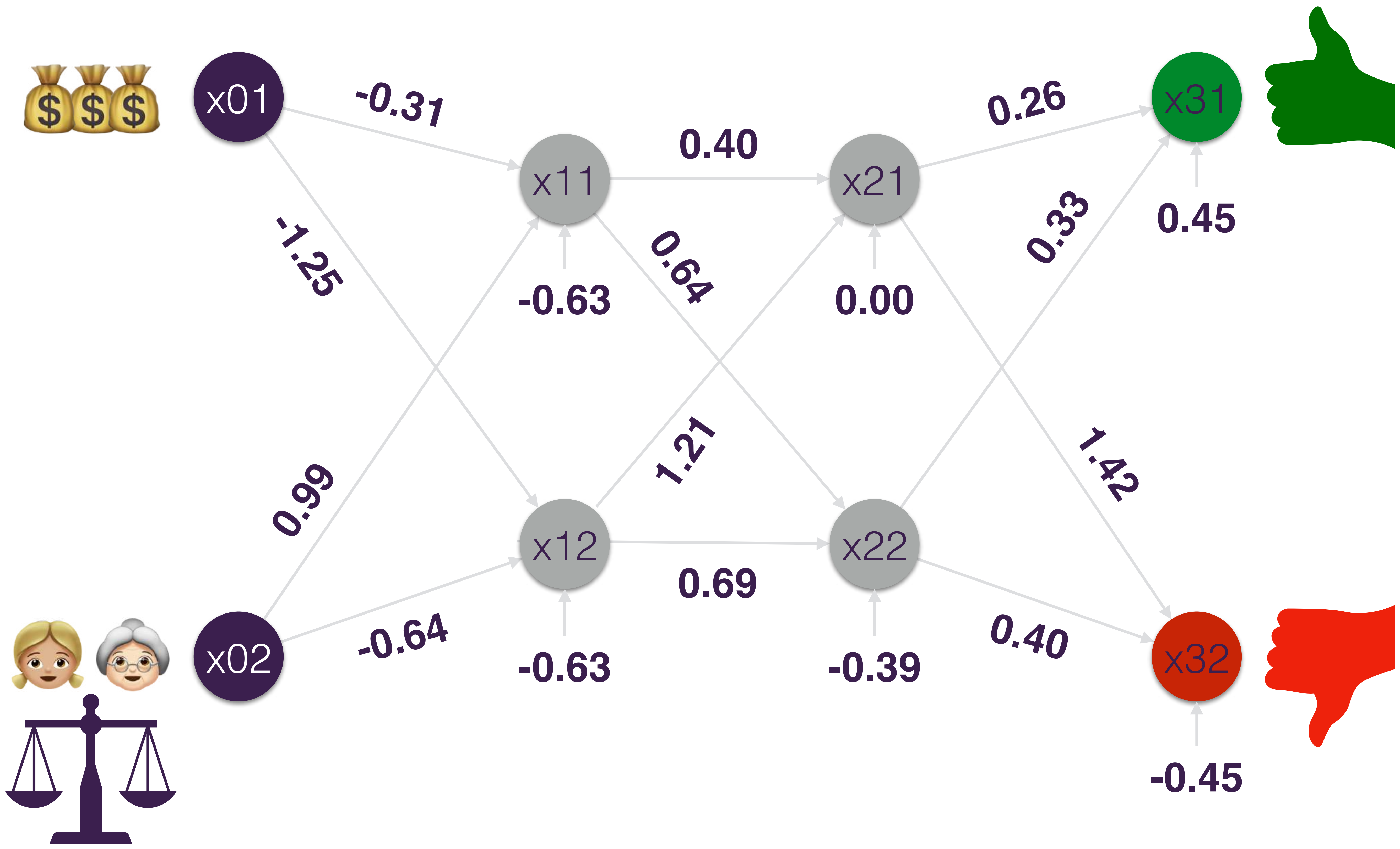


1. proceed **forwards** to find:
   - already ✓ fair partitions
   - activation patterns
2. proceed **backwards** for each activation pattern

Toy Example

Our Solution





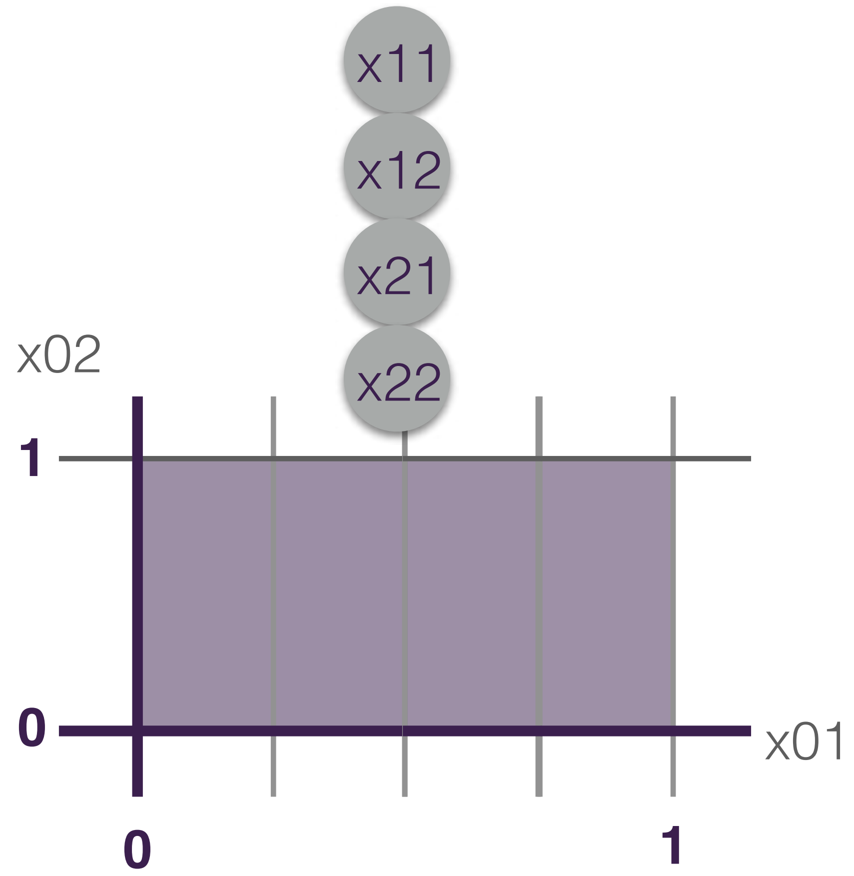
L = 0.25

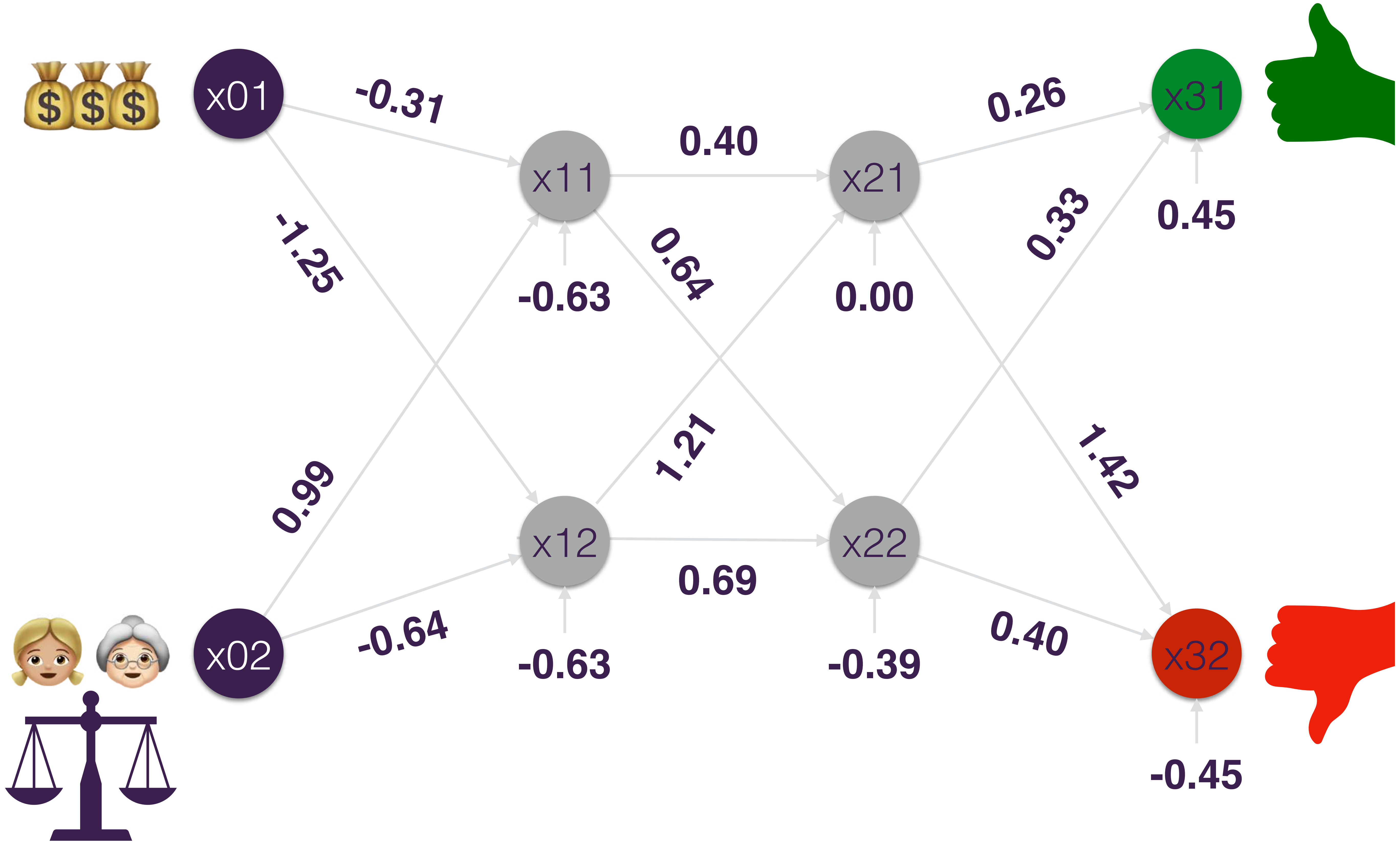
U = 2

```
x01 = input()
x02 = input()

x11 = -0.31 * x01 + 0.99 * x02 + (-0.63)
x12 = -1.25 * x01 + (-0.64) * x02 + 1.88

x11 = 0 if x11 < 0 else x11
x12 = 0 if x12 < 0 else x12

x21 = 0.40 * x11 + 1.21 * x12 + 0.00
x22 = 0.64 * x11 + 0.69 * x12 + (-0.39)

x21 = 0 if x21 < 0 else x21
x22 = 0 if x22 < 0 else x22

x31 = 0.26 * x21 + 0.33 * x22 + 0.45
x32 = 1.42 * x21 + 0.40 * x22 + (-0.45)

if x31 > x32:
   print('credit approved')
elif x32 < x31:
   print('credit denied')
```

Toy Example

Our Solution





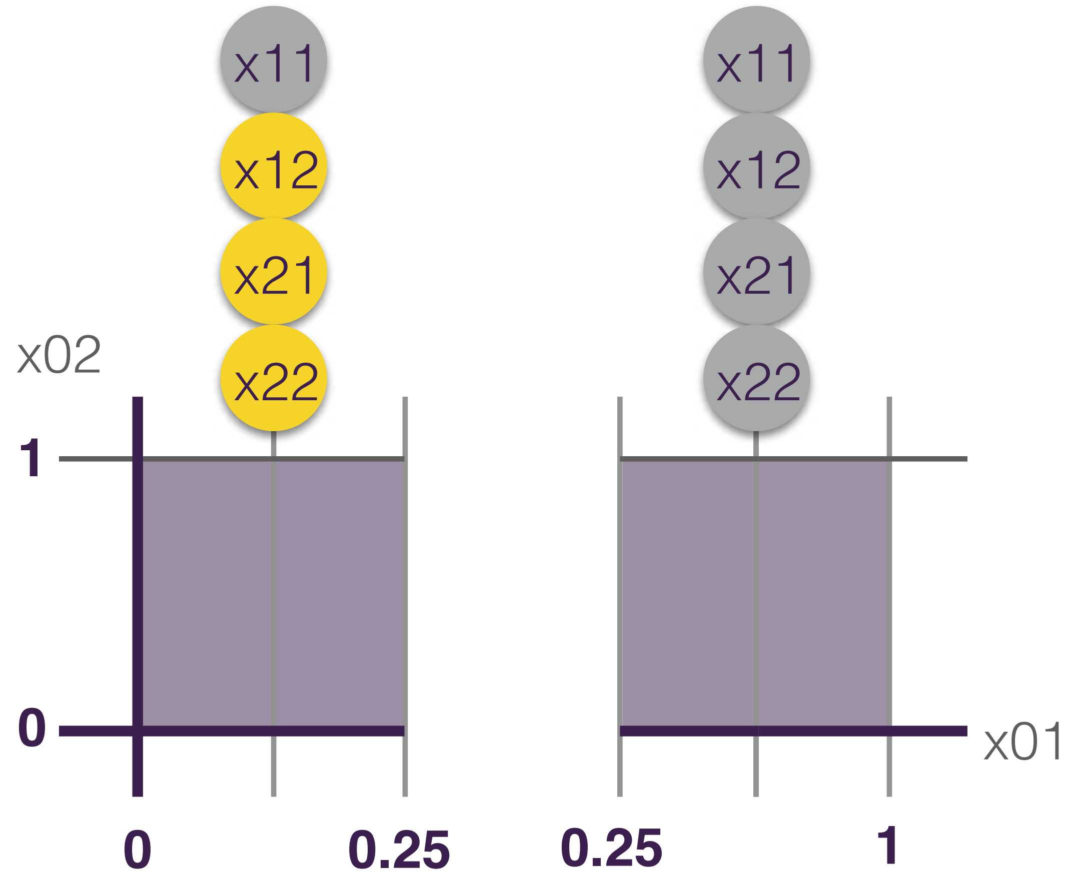
L = 0.25

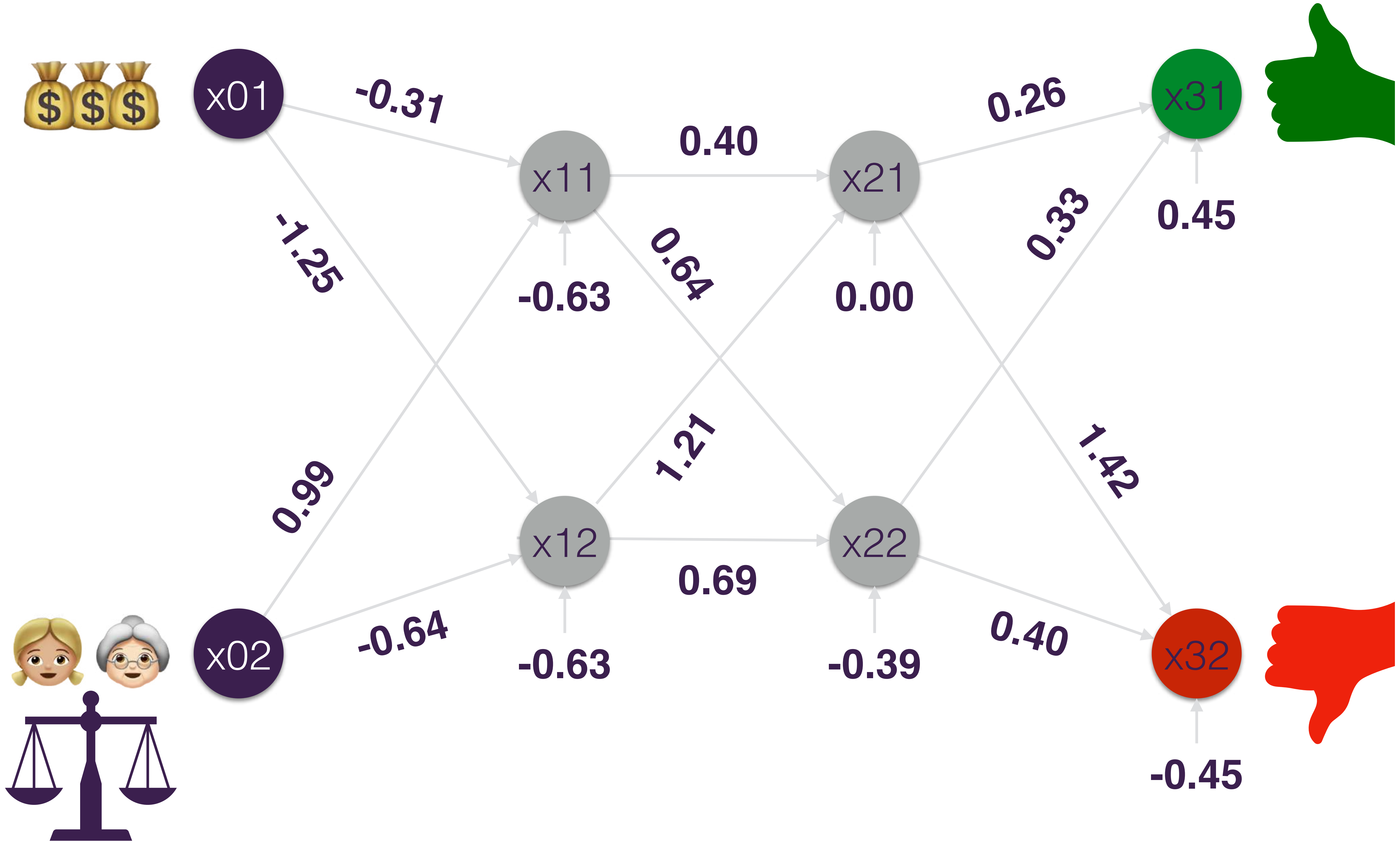
U = 2

```
x01 = input()
x02 = input()

x11 = -0.31 * x01 + 0.99 * x02 + (-0.63)
x12 = -1.25 * x01 + (-0.64) * x02 + 1.88

x11 = 0 if x11 < 0 else x11
x12 = 0 if x12 < 0 else x12

x21 = 0.40 * x11 + 1.21 * x12 + 0.00
x22 = 0.64 * x11 + 0.69 * x12 + (-0.39)

x21 = 0 if x21 < 0 else x21
x22 = 0 if x22 < 0 else x22

x31 = 0.26 * x21 + 0.33 * x22 + 0.45
x32 = 1.42 * x21 + 0.40 * x22 + (-0.45)

if x31 > x32:
   print('credit approved')
elif x32 < x31:
   print('credit denied')
```

Toy Example

Our Solution





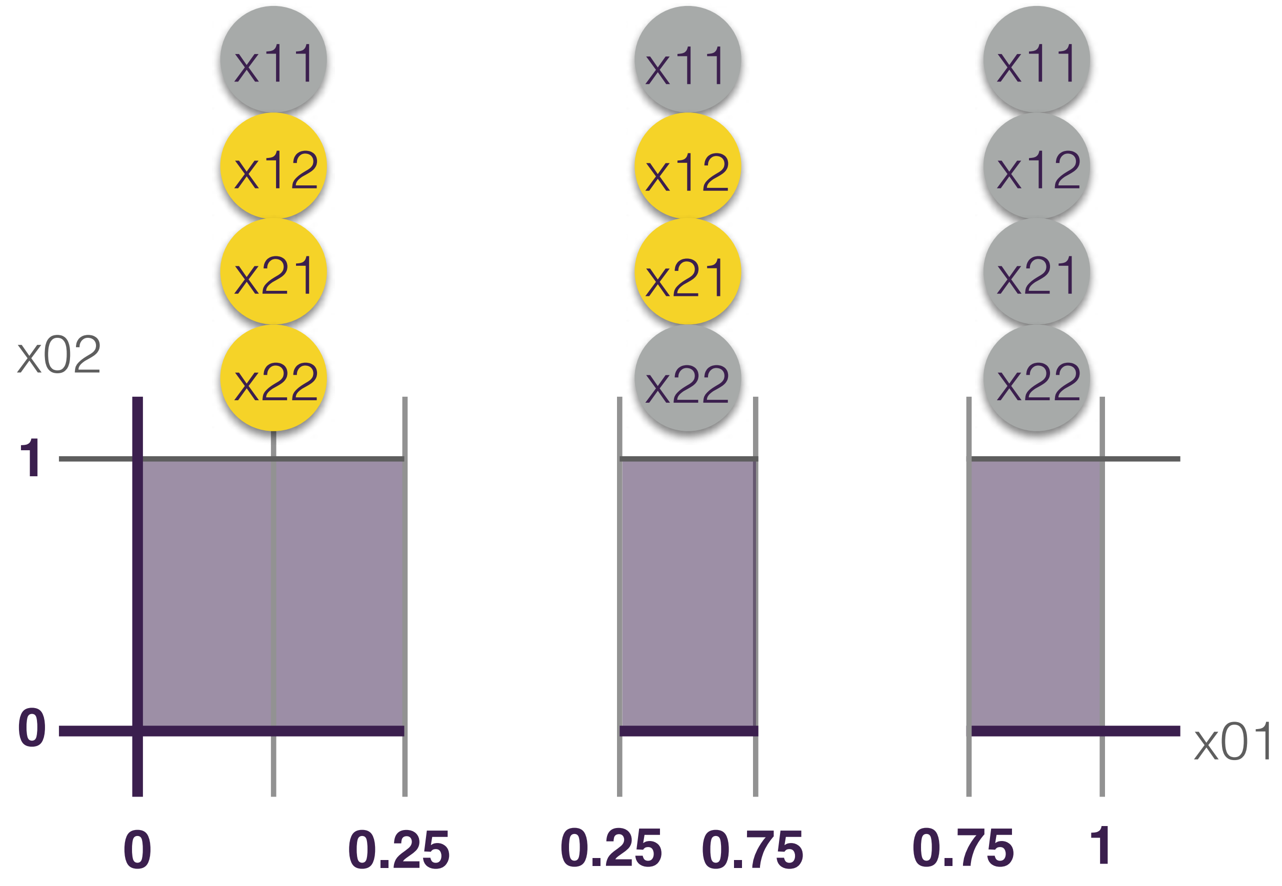
L = 0.25

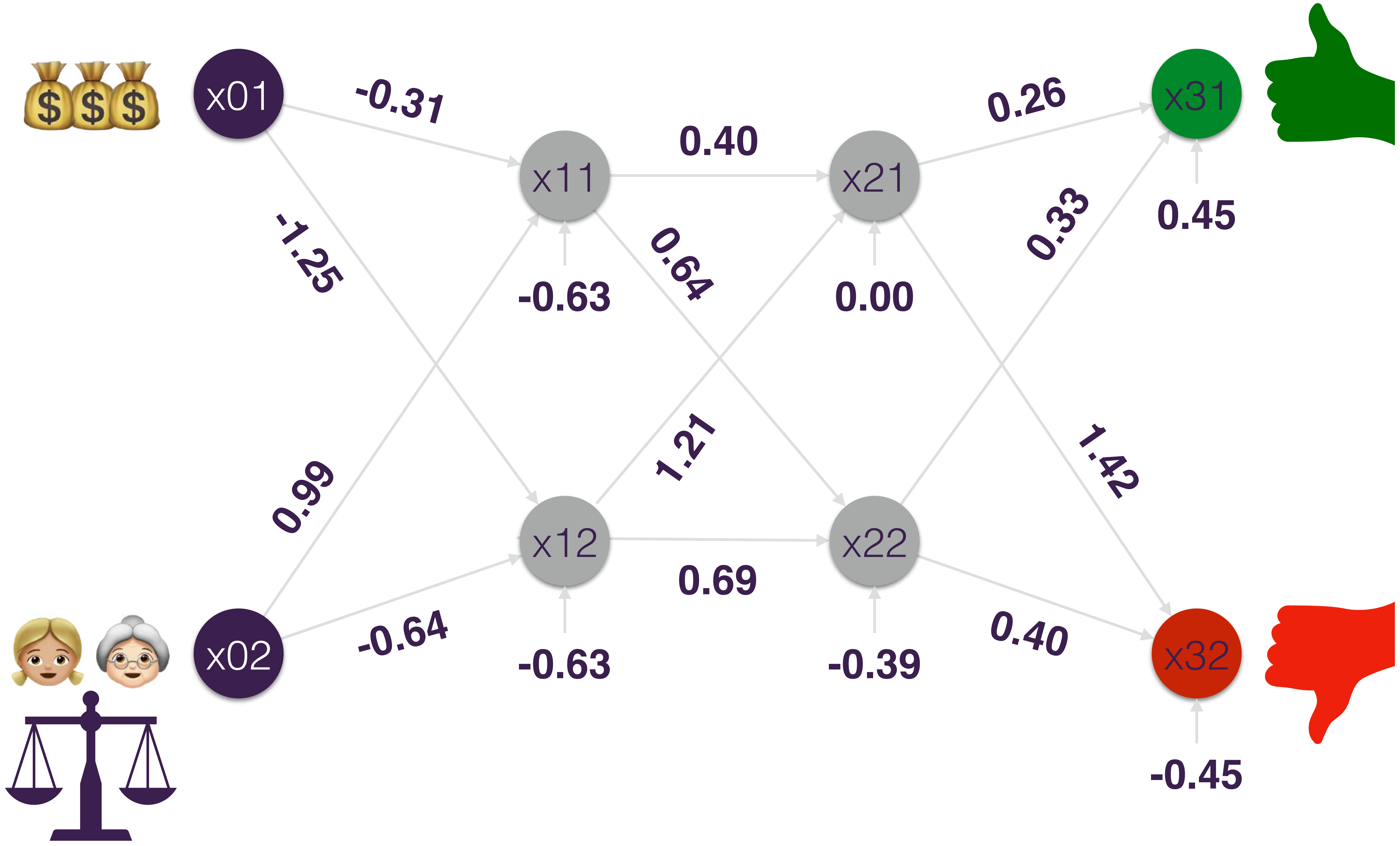
U = 2

```
x01 = input()
x02 = input()

x11 = -0.31 * x01 + 0.99 * x02 + (-0.63)
x12 = -1.25 * x01 + (-0.64) * x02 + 1.88

x11 = 0 if x11 < 0 else x11
x12 = 0 if x12 < 0 else x12

x21 = 0.40 * x11 + 1.21 * x12 + 0.00
x22 = 0.64 * x11 + 0.69 * x12 + (-0.39)

x21 = 0 if x21 < 0 else x21
x22 = 0 if x22 < 0 else x22

x31 = 0.26 * x21 + 0.33 * x22 + 0.45
x32 = 1.42 * x21 + 0.40 * x22 + (-0.45)

if x31 > x32:
   print('credit approved')
elif x32 < x31:
   print('credit denied')
```

Toy Example

Our Solution





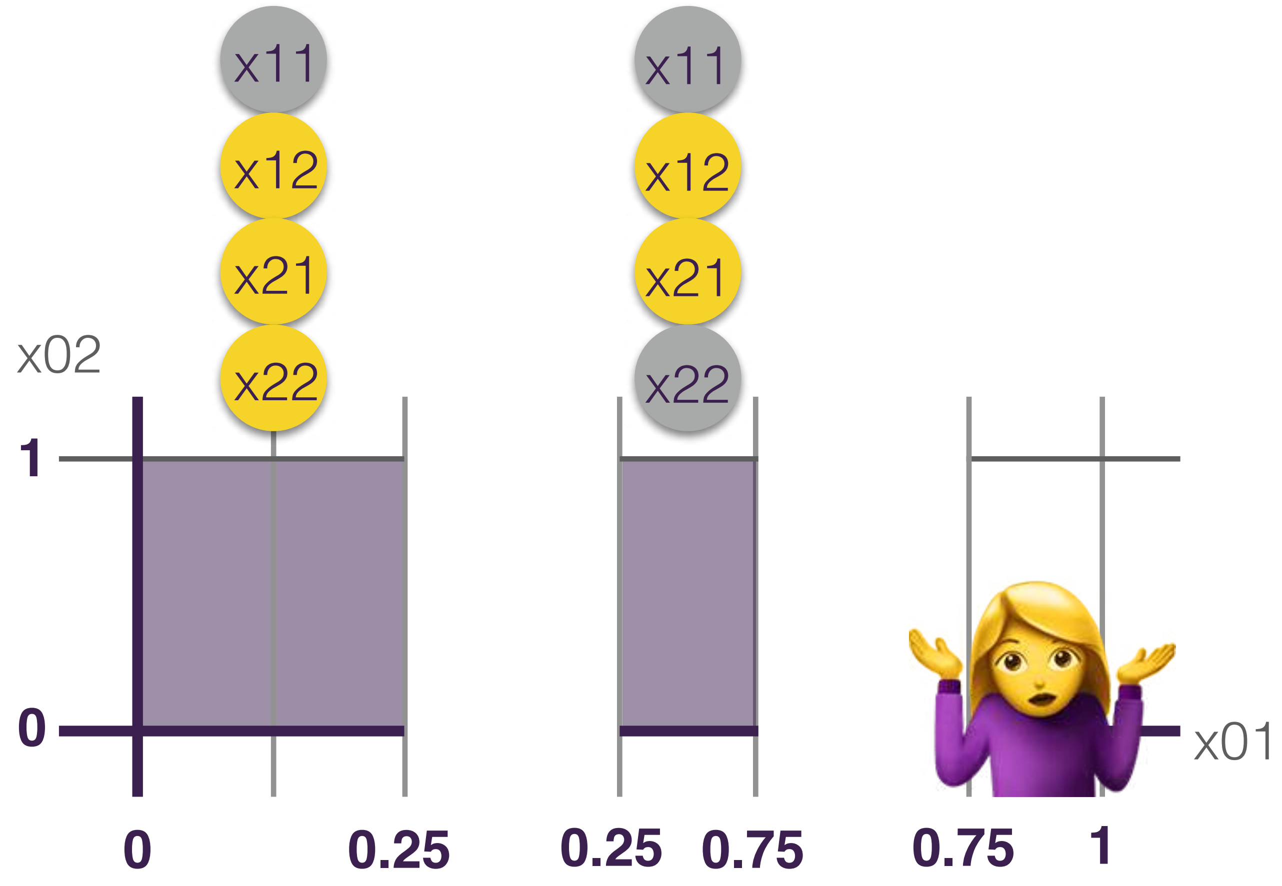
L = 0.25

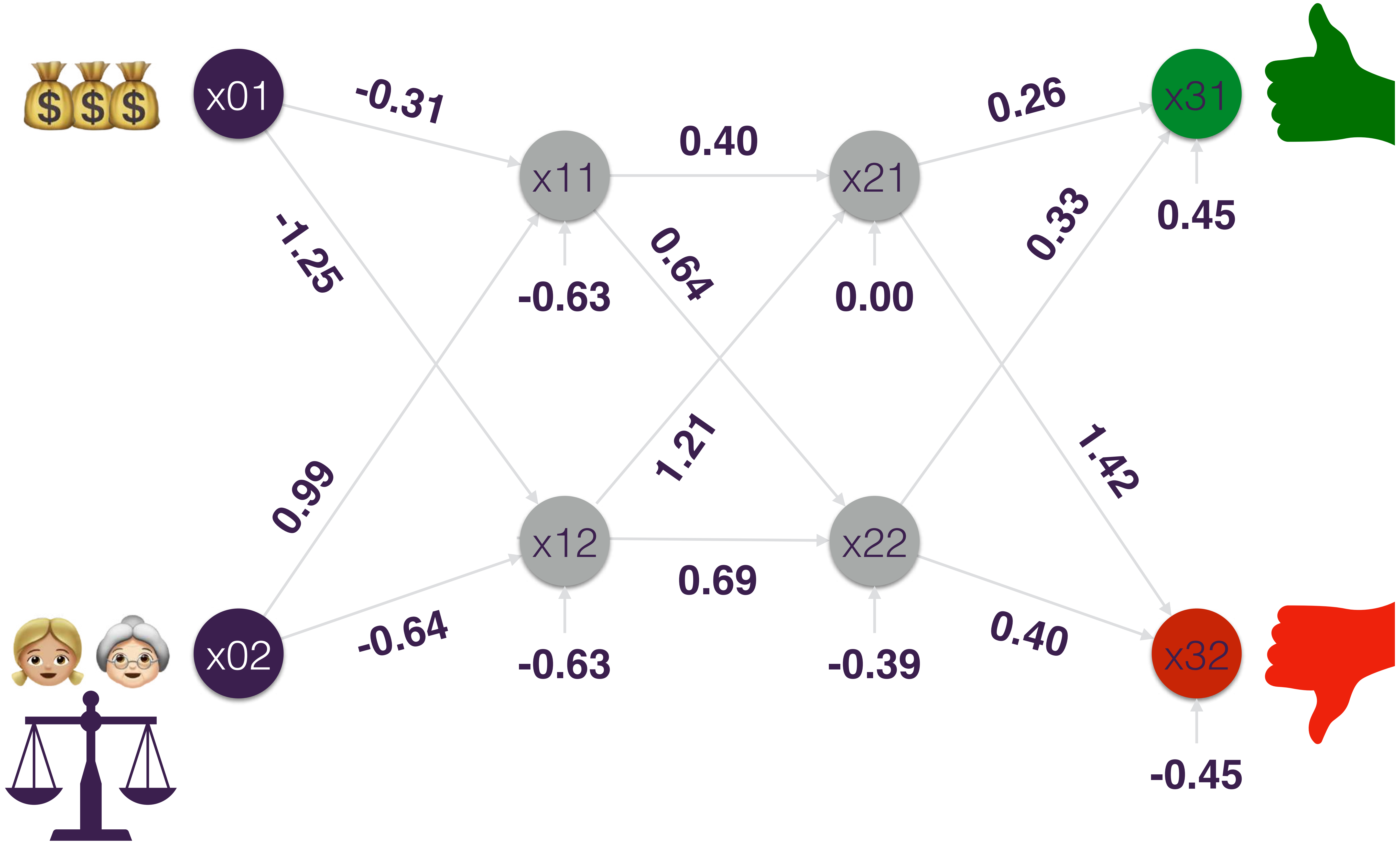
U = 2

```
x01 = input()
x02 = input()

x11 = -0.31 * x01 + 0.99 * x02 + (-0.63)
x12 = -1.25 * x01 + (-0.64) * x02 + 1.88

x11 = 0 if x11 < 0 else x11
x12 = 0 if x12 < 0 else x12

x21 = 0.40 * x11 + 1.21 * x12 + 0.00
x22 = 0.64 * x11 + 0.69 * x12 + (-0.39)

x21 = 0 if x21 < 0 else x21
x22 = 0 if x22 < 0 else x22

x31 = 0.26 * x21 + 0.33 * x22 + 0.45
x32 = 1.42 * x21 + 0.40 * x22 + (-0.45)

if x31 > x32:
   print('credit approved')
elif x32 < x31:
   print('credit denied')
```

Toy Example

Our Solution





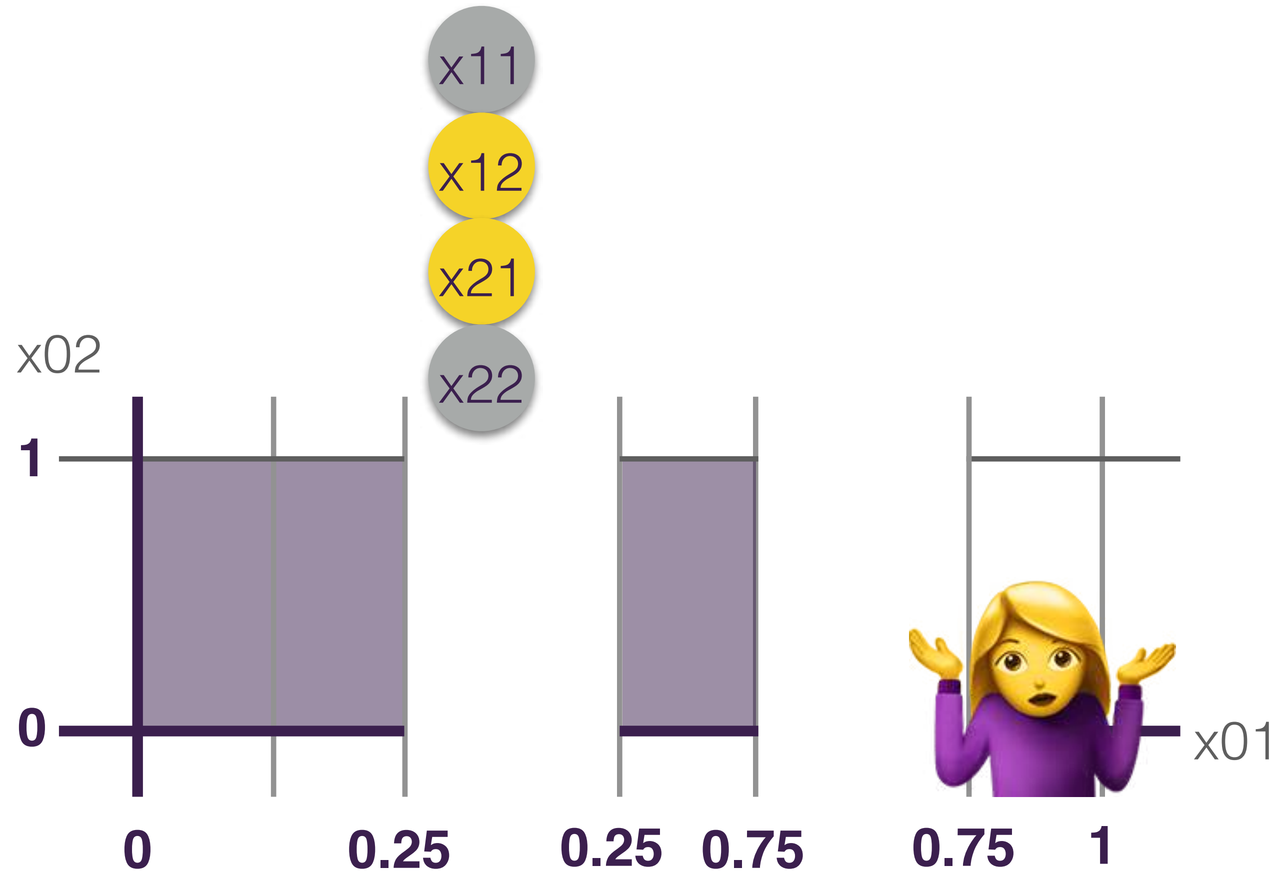
L = 0.25

U = 2

```
x01 = input()
x02 = input()

x11 = -0.31 * x01 + 0.99 * x02 + (-0.63)
x12 = -1.25 * x01 + (-0.64) * x02 + 1.88

x11 = 0 if x11 < 0 else x11
x12 = 0 if x12 < 0 else x12

x21 = 0.40 * x11 + 1.21 * x12 + 0.00
x22 = 0.64 * x11 + 0.69 * x12 + (-0.39)

x21 = 0 if x21 < 0 else x21
x22 = 0 if x22 < 0 else x22

x31 = 0.26 * x21 + 0.33 * x22 + 0.45
x32 = 1.42 * x21 + 0.40 * x22 + (-0.45)

if x31 > x32:
   print('credit approved')
elif x32 < x31:
   print('credit denied')
```

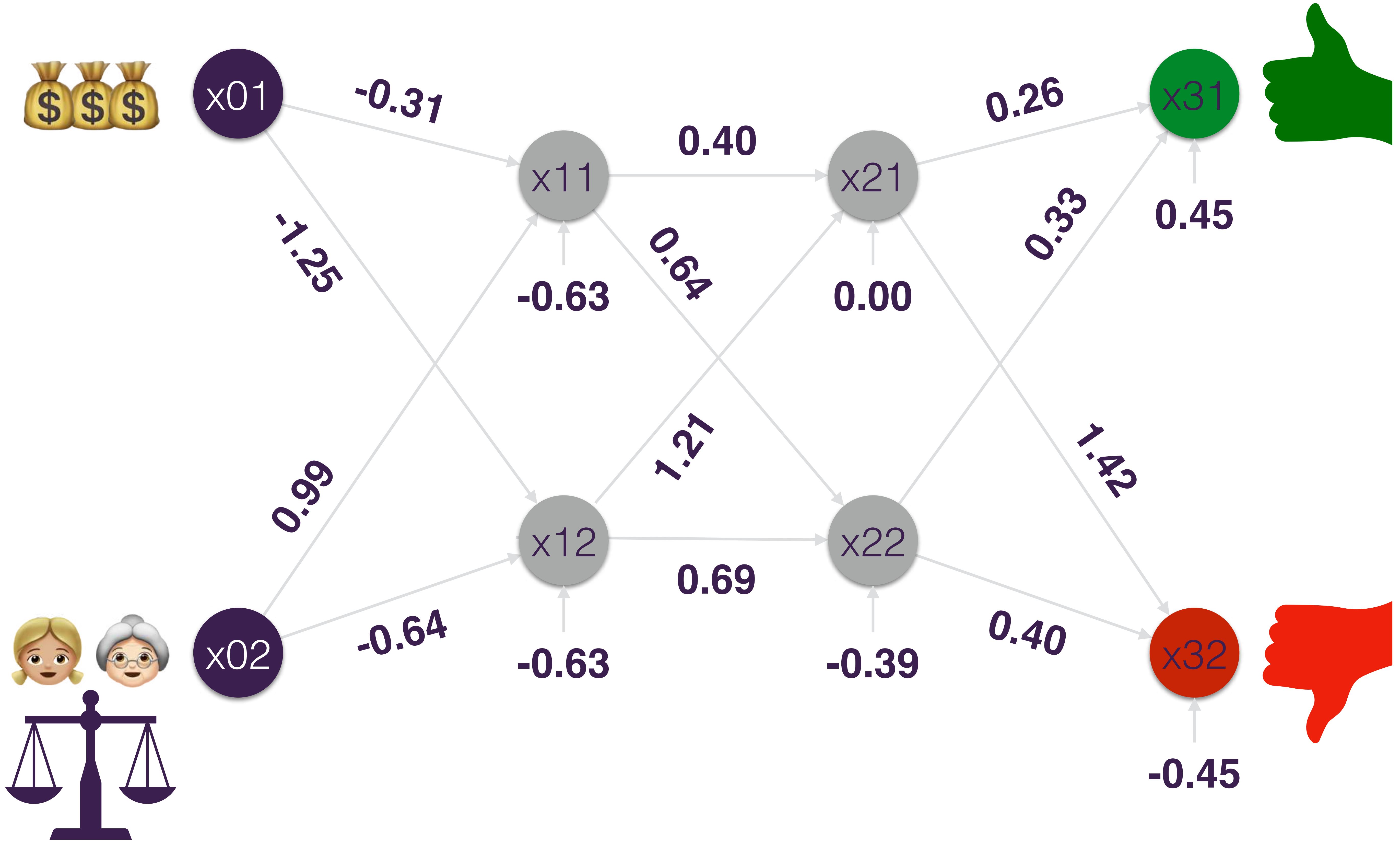
Toy Example

Our Solution



L = 0.25

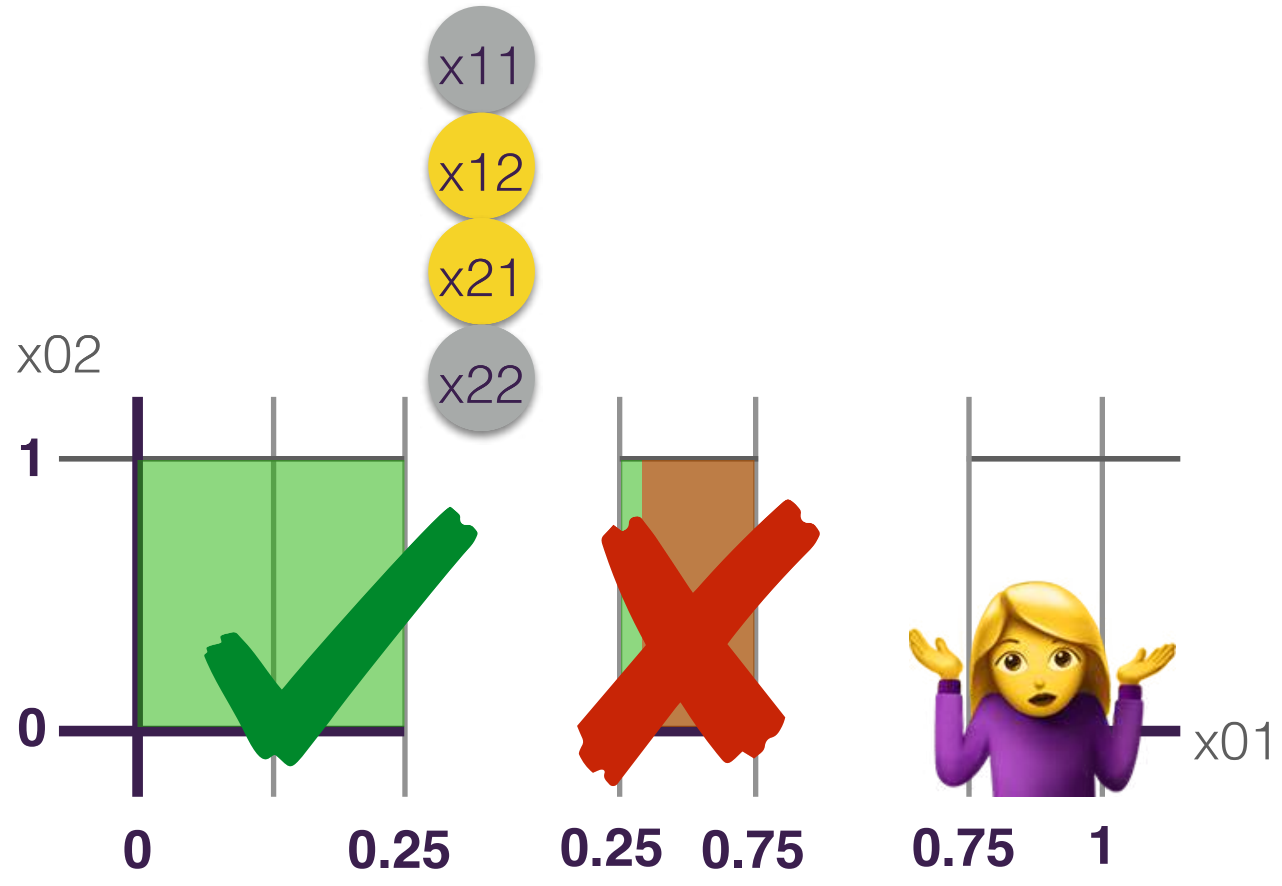
U = 2

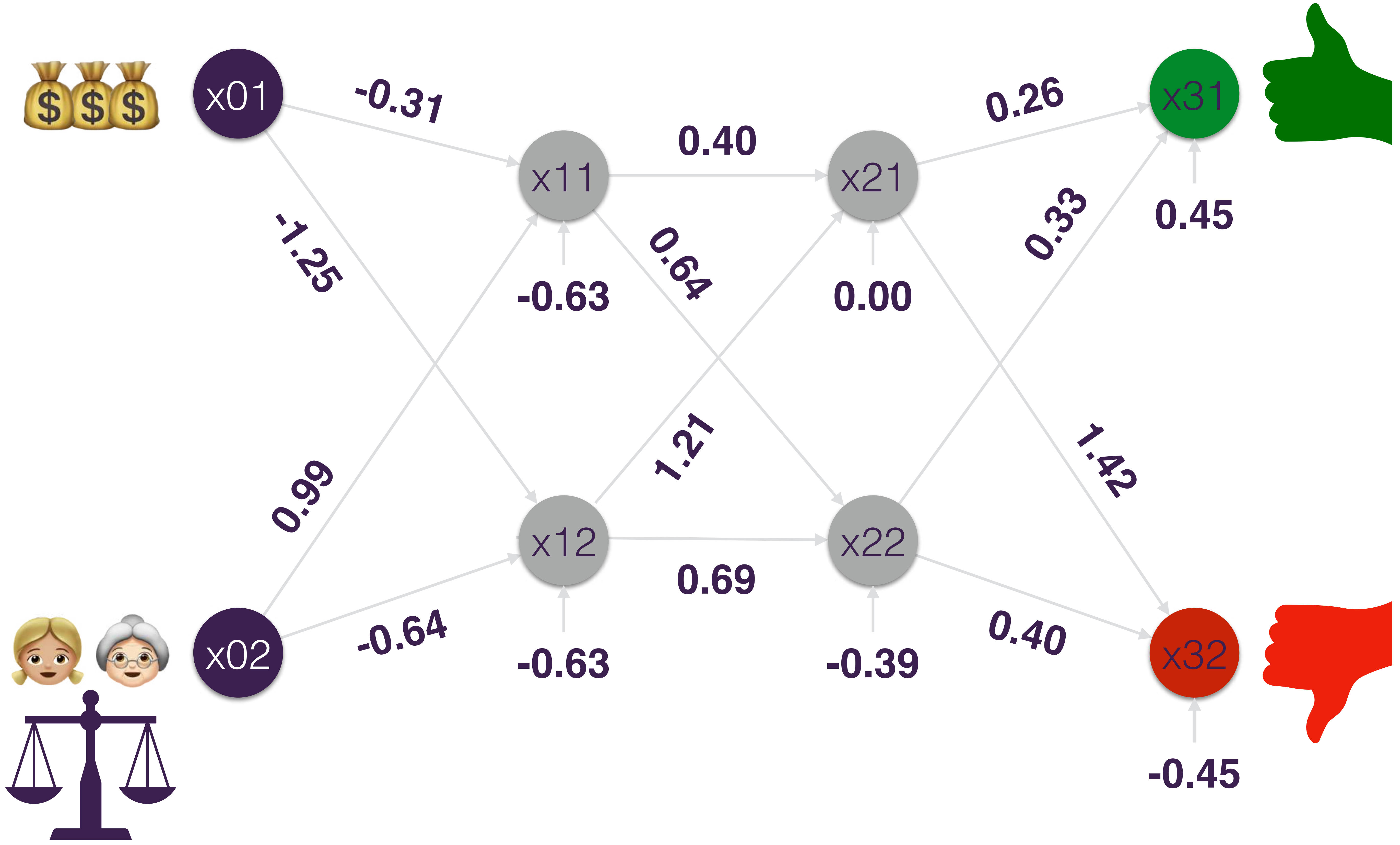



```
x01 = input()
x02 = input()

x11 = -0.31 * x01 + 0.99 * x02 + (-0.63)
x12 = -1.25 * x01 + (-0.64) * x02 + 1.88

x11 = 0 if x11 < 0 else x11
x12 = 0 if x12 < 0 else x12

x21 = 0.40 * x11 + 1.21 * x12 + 0.00
x22 = 0.64 * x11 + 0.69 * x12 + (-0.39)

x21 = 0 if x21 < 0 else x21
x22 = 0 if x22 < 0 else x22

x31 = 0.26 * x21 + 0.33 * x22 + 0.45
x32 = 1.42 * x21 + 0.40 * x22 + (-0.45)

if x31 > x32:
   print('credit approved')
elif x32 < x31:
   print('credit denied')
```

Toy Example

Our Solution






```
x01 = input()
x02 = input()
x11 = -0.31 * x
x12 = -1.25 * x
x11 = 0 if x11 < 0 else x11
x12 = 0 if x12 < 0 else x12
x21 = 0.40 * x11 + 1.21 * x12 + 0.00
x22 = 0.64 * x11 + 0.69 * x12 + (-0.39)
x21 = 0 if x21 < 0 else x21
x22 = 0 if x22 < 0 else x22
1.16 * x21 + 0.07 * x22 < 0.90    1.16 * x21 + 0.07 * x22 > 0.90
x31 = 0.26 * x21 + 0.33 * x22 + 0.45
x32 = 1.42 * x21 + 0.40 * x22 + (-0.45)
x31 > x32    x32 > x31
if x31 > x32:
   print('credit approved')
elif x32 < x31:
   print('credit denied')
```

check out the paper for the **formalization** and **soundness** proof!

check out our **artifact** for the implementation!

Scalability-vs-Precision Tradeoff

Japanese Credit Screening Dataset

L	U	⬟ BOXES					▲ SYMBOLIC									
		INPUT	|C|	|F|		TIME	INPUT	|C|	|F|		TIME	INPUT				
0.5	4	15.28%	37	0	0	8s	58.33%	79	8	20	1m 26s	69.79%				
	6	17.01%	39	6	6	51s	69.10%	129	22	61	5m 41s	80.56%	104	23	51	7m 53s
	8	51.39%	90	28	85	12m 2s	82.64%	88	31	67	12m 35s	91.32%	84	27	56	19m 33s
	10	79.86%	89	34	89	34m 15s	93.06%	98	40	83	42m 32s	96.88%	83	29	58	43m 39s
0.25	4	59.09%	1115	20	415	54m 32s	95.94%	884	39	484	54m 31s	98.26%	540	65	293	14m 29s
	6	83.77%	1404	79	944	37m 19s	98.68%	634	66	376	23m 31s	99.70%	322	79	205	13m 25s
	8	96.07%	869	140	761	1h 7m 29s	99.72%	310	67	247	1h 3m 33s	99.98%	247	69	177	22m 52s
	10	99.54%	409	93	403	1h 35m 20s	99.98%	195	52	176	1h 2m 13s	100.00%	111	47	87	34m 56s
0.125	4	97.13%	12449	200	9519	3h 33m 48s	99.99%	1101	60	685	47m 46s	99.99%	768	81	415	19m 1s
	6	99.83%	5919	276	4460	3h 23m	100.00%	988	77	606	26m 47s	100.00%	489	80	298	16m 54s
	8	99.98%	1926	203	1568	2h 14m 25s	100.00%	404	73	309	46m 31s	100.00%	175	57	129	20m 11s
	10	100.00%	428	95	427	1h 39m 31s	100.00%	151	53	141	57m 32s	100.00%	80	39	62	28m 33s
0	4	100.00%	19299	295	15446	6h 13m 24s	100.00%	1397	60	885	40m 5s	100.00%	766	87	425	**16m 41s**
	6	100.00%	4843	280	3679	2h 24m 7s	100.00%	763	66	446	35m 24s	100.00%	401	81	242	32m 29s
	8	100.00%	1919	208	1567	2h 9m 59s	100.00%	404	73	309	45m 48s	100.00%	193	68	144	24m 16s
	10	100.00%	486	102	475	1h 41m 3s	100.00%	217	55	192	1h 2m 11s	100.00%	121	50	91	30m 53s

- a larger U or a smaller L improves **precision**
- a more precise forward analysis improves **scalability**

https://archive.ics.uci.edu/ml/datasets/Japanese+Credit+Screening

Seeded Bias and Bias Queries

German Credit and ProPublica COMPAS Datasets

CREDIT	BOXES				SYMBOLIC				DEEPPOLY				
	FAIR DATA		BIASED DATA		FAIR DATA		BIASED DATA		FAIR DATA		BIASED DA[illegible]		
	BIAS	TIME	BIAS	TIME	BIAS	TIME	BIAS	TIME	BIAS	TIME	BIAS	T[illegible]	
≤ 1000	0.09%	47s	0.09%	2m 17s	0.09%	13s	0.09%	1m 10s	0.09%	**10s**	0.09%	[illegible]	[illegible]
	0.19%	5m 46s	0.45%	13m 2s	0.19%	**1m 5s**	0.45%	2m 41s	0.19%	1m 12s	0.45%	**1m 46s**	MEDIAN
	0.33%	30m 59s	0.95%	1h 56m 57s	0.33%	**4m 8s**	0.95%	**13m 16s**	0.33%	5m 45s	0.95%	18m 18s	MAX
> 1000	2.21%	1m 42s	4.52%	21m 11s	2.21%	**38s**	4.52%	**3m 7s**	2.21%	39s	4.52%	4m 44s	MIN
	6.72%	31m 42s	23.41%	1h 36m 51s	6.72%	8m 59s	23.41%	41m 44s	6.63%	**4m 58s**	23.41%	**15m 39s**	MEDIAN
	14.96%	7h 7m 12s	33.19%	16h 50m 48s	14.96%	4h 16m 52s	33.19%	8h 5m 14s	14.96%	**1h 9m 45s**	31.17%	**6h 51m 50s**	MAX

QUERY	BOXES				SYMBOLIC				DEEPPOLY				
	FAIR DATA		BIASED DATA		FAIR DATA		BIASED DATA		FAIR DATA		BIASED DATA		
	BIAS	TIME	BIAS	TIME	BIAS	TIME	BIAS	TIME	BIAS	TIME	BIAS	TIME	
AGE < 25 RACE BIAS?	0.22%	24m 32s	0.12%	14m 53s	0.22%	11m 34s	0.12%	**7m 14s**	0.22%	**5m 18s**	0.12%	8m 46s	MIN
	0.31%	1h 54m 48s	0.99%	57m 33s	0.32%	**36m 0s**	0.99%	20m 43s	0.32%	47m 16s	0.99%	**16m 38s**	MEDIAN
	2.46%	2h 44m 11s	8.33%	5h 29m 19s	2.46%	2h 17m 3s	8.50%	3h 34m 50s	2.12%	**1h 11m 43s**	6.48%	**2h 5m 5s**	MAX
MALE AGE BIAS?	2.60%	24m 14s	4.51%	34m 23s	2.64%	25m 13s	5.20%	29m 19s	2.70%	**19m 47s**	5.22%	**20m 51s**	MIN
	6.08%	1h 49m 42s	6.95%	2h 3m 39s	6.77%	**1h 1m 51s**	7.02%	1h 2m 26s	6.77%	1h 13m 31s	7.00%	**47m 28s**	MEDIAN
	8.00%	5h 56m 6s	12.56%	8h 26m 55s	8.40%	**2h 2m 22s**	12.71%	4h 55m 35s	8.84%	2h 20m 23s	12.88%	**3h 25m 21s**	MAX
CAUCASIAN PRIORS BIAS?	2.18%	2h 54m 18s	2.92%	46m 53s	2.18%	1h 20m 41s	2.92%	30m 23s	2.18%	**18m 26s**	2.92%	**15m 29s**	MIN
	2.95%	6h 56m 44s	4.21%	3h 50m 38s	2.95%	4h 12m 28s	4.21%	3h 32m 52s	2.95%	**2h 36m 1s**	4.21%	**1h 34m 7s**	MEDIAN
	5.36%	**45h 2m 12s**	6.98%	70h 50m 10s	5.36%	60h 53m 6s	6.98%	49h 51m 42s	5.36%	52h 10m 2s	6.95%	**17h 48m 22s**	MAX

- our approach can effectively detect bias
- our approach can answer bias queries

https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis

Scalability wrt Neural Network Size

Adult Census Dataset

|M|	U	BOXES INPUT	BOXES |C|	BOXES |F|		BOXES TIME	SYMBOLIC INPUT	SYMBOLIC |C|	SYMBOLIC |F|		SYMBOLIC TIME	DEEP INPUT	DEEP |C|	DEEP |F|		DEEP TIME
10 ○ ● ⊕	4	88.26%	1482	77	1136	33m 55s	95.14%	1132	65	686	19m 5s	93.99%	1894	77	[illegible]	[illegible]
	6	99.51%	769	51	723	1h 10m 25s	99.93%	578	47	447	39m 8s	99.83%	1620	54	[illegible]	[illegible]
	8	100.00%	152	19	143	3h 47m 23s	100.00%	174	18	146	1h 51m 2s	100.00%	1170	26	824	8h 2m 27s
	10	100.00%	1	1	1	**55m 58s**	100.00%	1	1	1	56m 8s	100.00%	1	1	1	56m 43s
12 △ ▲ ⅄	4	49.83%	719	9	329	13m 43s	72.29%	1177	11	559	24m 9s	60.52%	1498	14	423	10m 32s
	6	72.74%	1197	15	929	2h 6m 49s	98.54%	333	7	195	20m 46s	66.46%	1653	17	594	15m 44s
	8	98.68%	342	9	284	1h 46m 43s	98.78%	323	9	190	1h 27m 18s	70.87%	1764	18	724	2h 19m 11s
	10	99.06%	313	7	260	1h 21m 47s	99.06%	307	5	182	**1h 13m 55s**	80.76%	1639	18	1007	3h 22m 11s
20 ◇ ◆ ⬙	4	38.92%	1044	18	39	2m 6s	51.01%	933	31	92	15m 28s	49.62%	1081	34	79	3m 2s
	6	46.22%	1123	62	255	20m 51s	61.60%	916	67	405	44m 40s	59.20%	1335	90	356	22m 13s
	8	64.24%	1111	96	792	2h 24m 51s	74.27%	1125	78	780	3h 26m 20s	69.69%	1574	127	652	5h 6m 7s
	10	85.90%	1390	71	1339	>13h	89.27%	1435	60	1157	>13h	76.25%	1711	148	839	**4h 36m 23s**
40 □ ■ ⬙	4	0.35%	10	0	0	1m 39s	34.62%	768	1	1	6m 56s	26.39%	648	2	3	10m 11s
	6	0.35%	10	0	0	1m 38s	34.76%	817	4	5	43m 53s	26.74%	592	8	10	1h 23m 11s
	8	0.42%	12	1	2	14m 37s	35.56%	840	21	28	2h 48m 15s	27.74%	686	32	42	2h 43m 2s
	10	0.80%	23	10	13	1h 48m 43s	37.19%	880	50	75	**11h 32m 21s**	30.56%	699	83	121	>13h
45 ⬠ ⬟ ✳	4	1.74%	50	0	0	1m 38s	41.98%	891	14	49	10m 14s	36.60%	805	6	8	2m 47s
	6	2.50%	72	3	22	4m 35s	45.00%	822	32	143	45m 42s	38.06%	847	25	50	5m 7s
	8	9.83%	282	25	234	25m 30s	47.78%	651	46	229	1h 14m 5s	42.53%	975	74	180	25m 1s
	10	18.68%	522	33	488	1h 51m 24s	49.62%	714	51	294	**3h 23m 20s**	48.68%	1087	110	373	1h 58m 34s

- scalability degrades for larger neural networks (less for models with fewer nodes per layer)
- a larger U sometimes improves scalability

https://archive.ics.uci.edu/ml/datasets/adult

Scalability wrt Queried Input Space

Adult Census Dataset

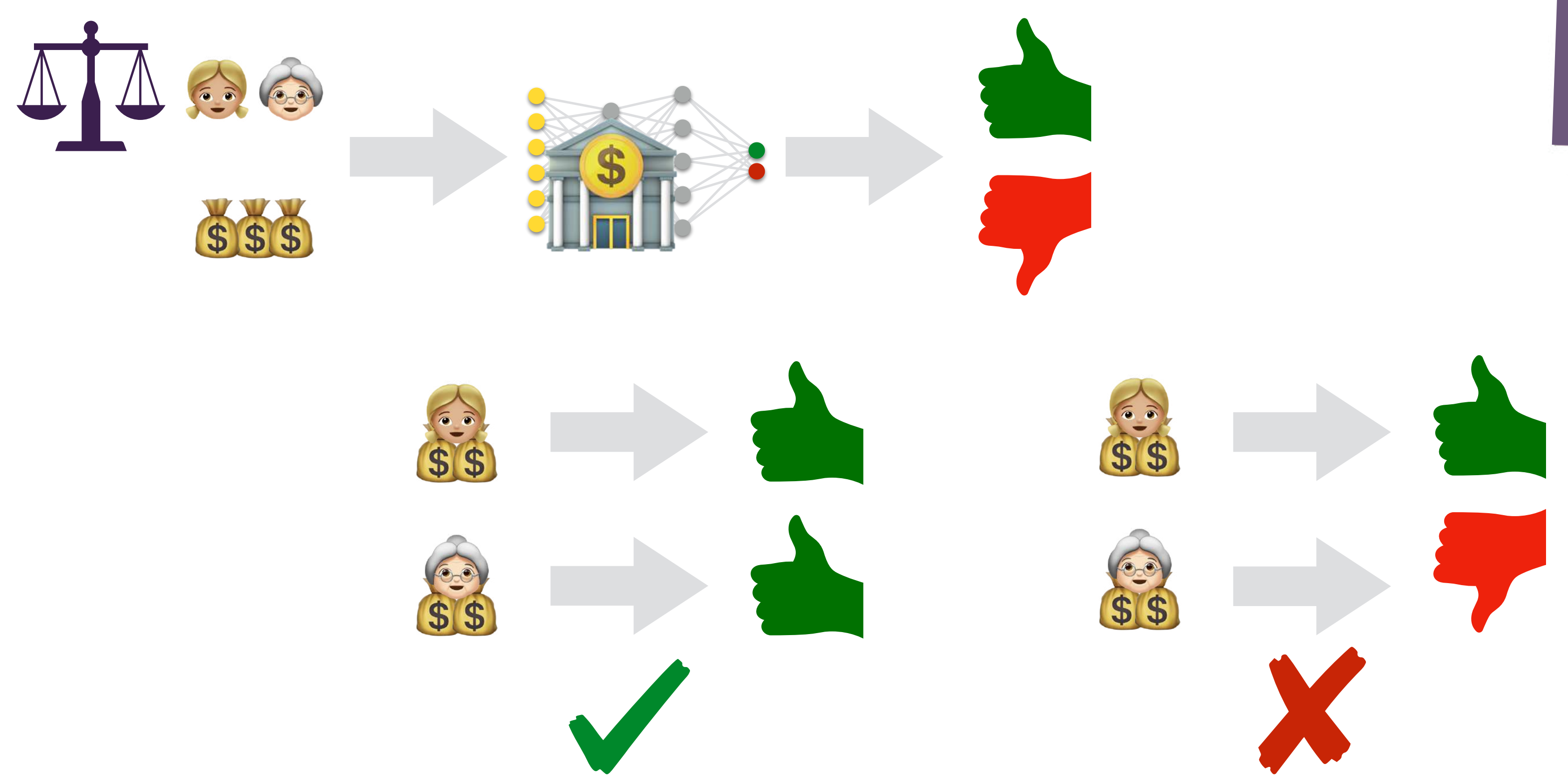
|M|	QUERY	BOXES INPUT	BOXES |C|	BOXES |F|		BOXES TIME	SYMBOLIC INPUT	SYMBOLIC |C|	SYMBOLIC |F|		SYMBOLIC TIME	DEEPPOLY INPUT	DEEPPOLY |C|	DEEPPOLY |F|		DEEPPOLY TIME
80	F 0.009%	99.931% 0.009%	11	0	0	3m 5s	99.961% 0.009%	17	0	0	**3m 2s**	99.957% 0.009%	10	0	0	2m 36s
	E 0.104%	99.583% 0.104%	61	0	0	3m 6s	99.783% 0.104%	89	0	0	**3m 10s**	99.753% 0.104%	74	0	0	2m 44s
	D 1.042%	97.917% 1.020%	151	0	0	2m 56s	99.258% 1.034%	297	0	0	**3m 41s**	98.984% 1.031%	477	0	0	2m 58s
	C 8.333%	83.503% 6.958%	506	2	3	2h 1m	95.482% 7.956%	885	25	34	>13h	93.225% 7.768%	1145	23	33	**12h 57m 37s**
	B 50%	25.634% 12.817%	5516	7	11	**1h 28m 6s**	76.563% 38.281%	4917	123	182	>13h	63.906% 31.953%	7139	117	152	>13h
	A 100%	0.052% 0.052%	12	0	0	25m 51s	61.385% 61.385%	5156	73	102	**10h 25m 2s**	43.698% 43.698%	4757	68	88	>13h
320	F 0.009%	99.931% 0.009%	6	0	0	3m 15s	99.944% 0.009%	9	0	0	**3m 35s**	99.931% 0.009%	6	0	0	3m 30s
	E 0.104%	99.583% 0.104%	121	0	0	3m 39s	99.627% 0.104%	120	0	0	**6m 34s**	99.583% 0.104%	31	0	0	4m 22s
	D 1.042%	97.917% 1.020%	151	0	0	6m 18s	98.247% 1.024%	597	0	0	**21m 9s**	97.917% 1.020%	301	0	0	9m 35s
	C 8.333%	83.333% 6.944%	120	0	0	30m 37s	88.294% 7.358%	755	0	0	**1h 36m 35s**	83.342% 6.945%	483	0	0	52m 29s
	B 50%	25.000% 12.500%	5744	0	0	2h 24m 36s	46.063% 23.032%	4676	0	0	**7h 25m 57s**	25.074% 12.537%	5762	4	4	>13h
	A 100%	0.000% 0.000%	0	0	0	2h 54m 25s	24.258% 24.258%	2436	0	0	**9h 41m 36s**	0.017% 0.017%	4	0	0	5h 3m 33s
1280	F 0.009%	99.931% 0.009%	11	0	0	7m 35s	99.948% 0.009%	10	0	0	**24m 42s**	99.931% 0.009%	6	0	0	7m 6s
	E 0.104%	99.583% 0.104%	31	0	0	15m 49s	99.674% 0.104%	71	0	0	**51m 52s**	99.583% 0.104%	31	0	0	15m 14s
	D 1.042%	97.917% 1.020%	151	0	0	1h 49s	98.668% 1.028%	557	0	0	**3h 31m 45s**	97.917% 1.020%	301	0	0	1h 3m 33s
	C 8.333%	83.333% 6.944%	481	0	0	**7h 11m 39s**	–	–	–	–	>13h	83.333% 6.944%	481	0	0	7h 12m 57s
	B 50%	–	–	–	–	>13h	–	–	–	–	>13h	–	–	–	–	>13h
	A 100%	–	–	–	–	>13h	–	–	–	–	>13h	–	–	–	–	>13h

the **size of the queried input space** (rather than the size of the neural network) is the **most important factor** for scalability!

Dependency Fairness

the output classification is independent of the values of the sensitive input feature(s)

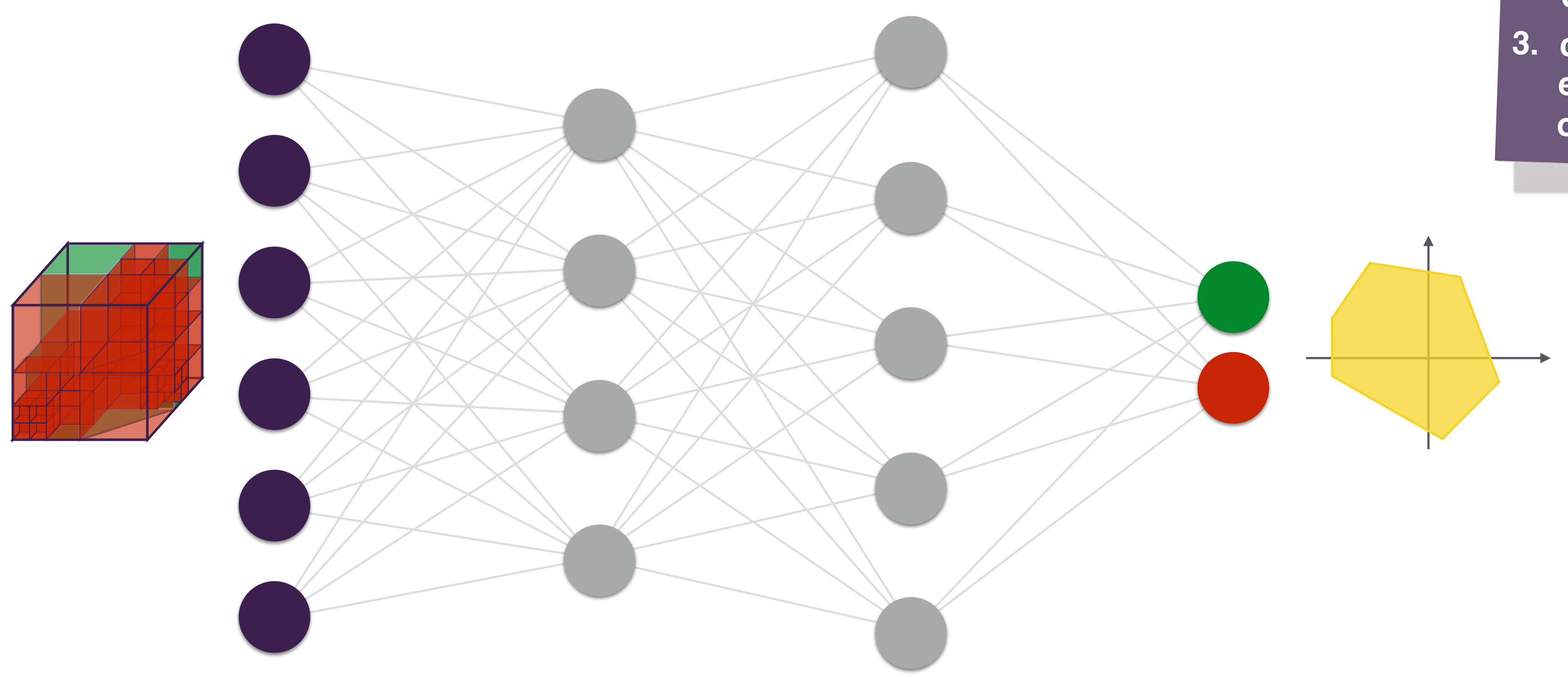
- does not require an oracle
- amenable to static analysis
- stronger than group fairness



Galhotra et al. - Fairness Testing: Testing Software for Discrimination (FSE 2017)

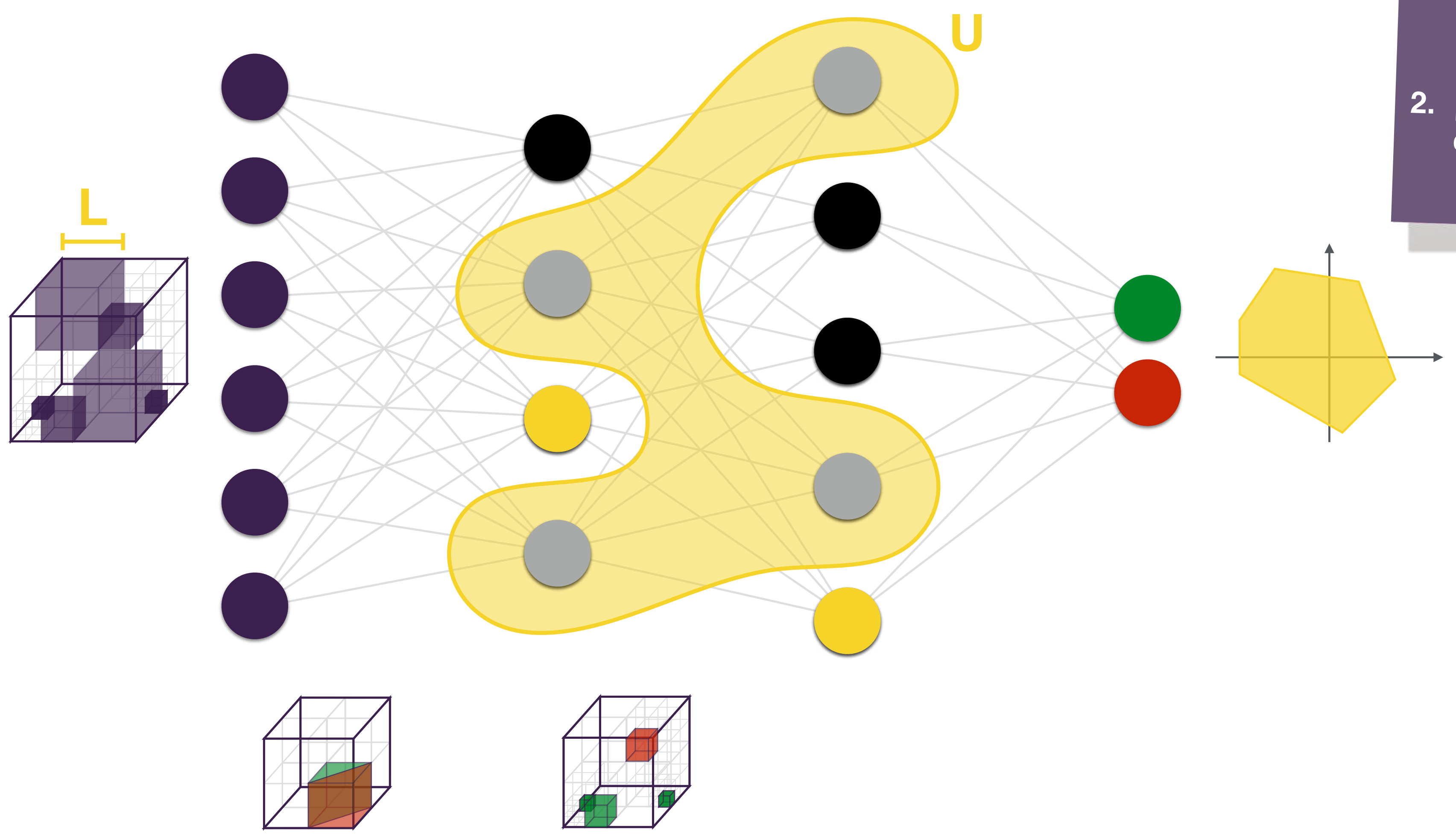
Naïve Backward Analysis

1. proceed backwards from all possible classifications
2. project away the value of the sensitive feature(s)
3. check for intersection:
   empty → fair
   otherwise → alarm



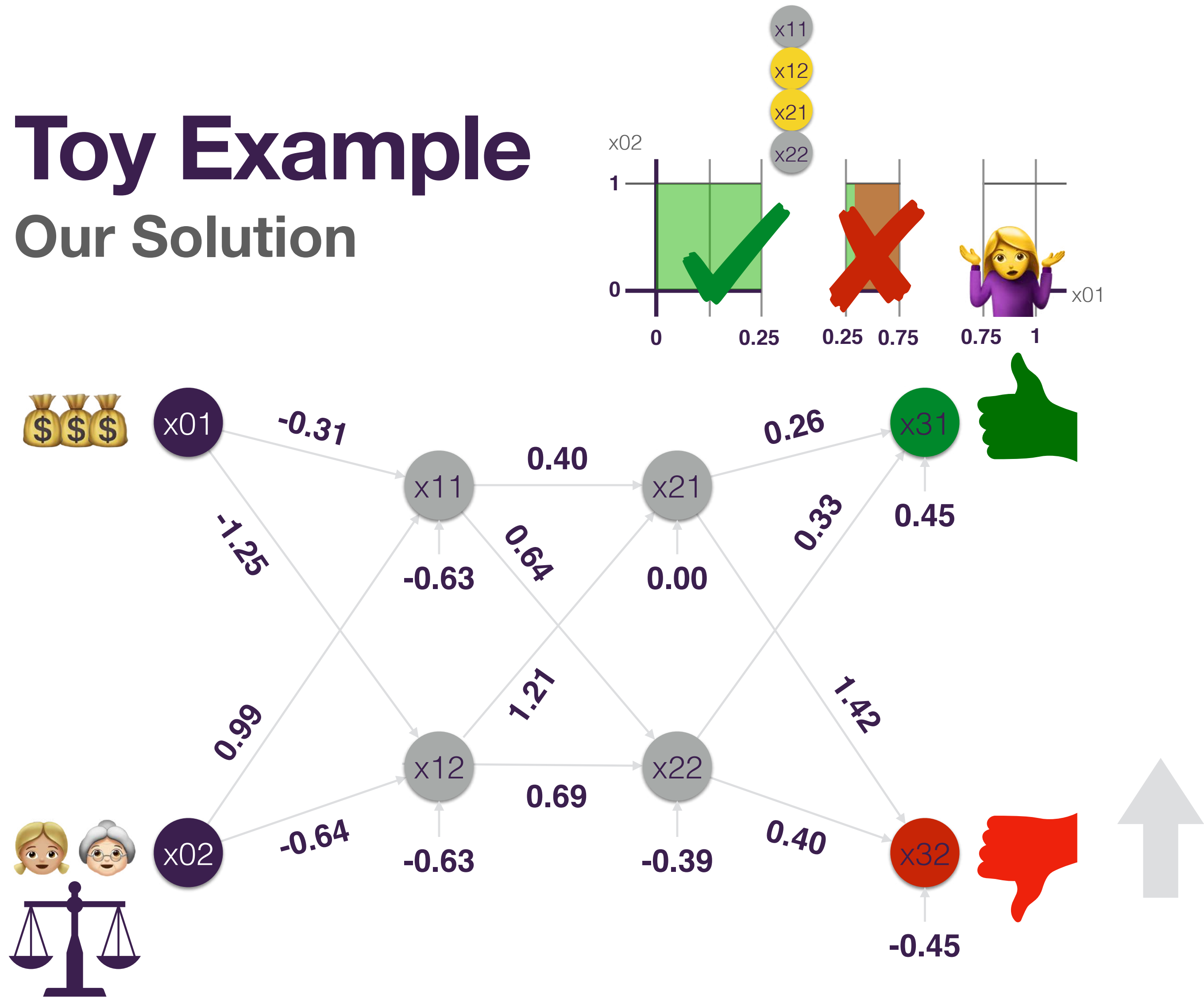
Our Solution

1. proceed forwards to find:
   - already fair partitions
   - activation patterns
2. proceed backwards for each activation pattern



Toy Example

Our Solution




```
x01 = input()
x02 = input()
x11 = -0.31 * x01 + 0.99 * x02 + (-0.63)
x12 = -1.25 * x01 + (-0.64) * x02 + 1.88
x11 = 0 if x11 < 0 else x11
x12 = 0 if x12 < 0 else x12
x21 = 0.40 * x11 + 1.21 * x12 + 0.00
x22 = 0.64 * x11 + 0.69 * x12 + (-0.39)
x21 = 0 if x21 < 0 else x21
x22 = 0 if x22 < 0 else x22
1.16 * x21 + 0.07 * x22 < 0.90    1.16 * x21 + 0.07 * x22 > 0.90
x31 = 0.26 * x21 + 0.33 * x22 + 0.45
x32 = 1.42 * x21 + 0.40 * x22 + (-0.45)
x31 > x32    x32 > x31
if x31 > x32:
    print('credit approved')
elif x32 < x31:
    print('credit denied')
```

QUESTIONS?